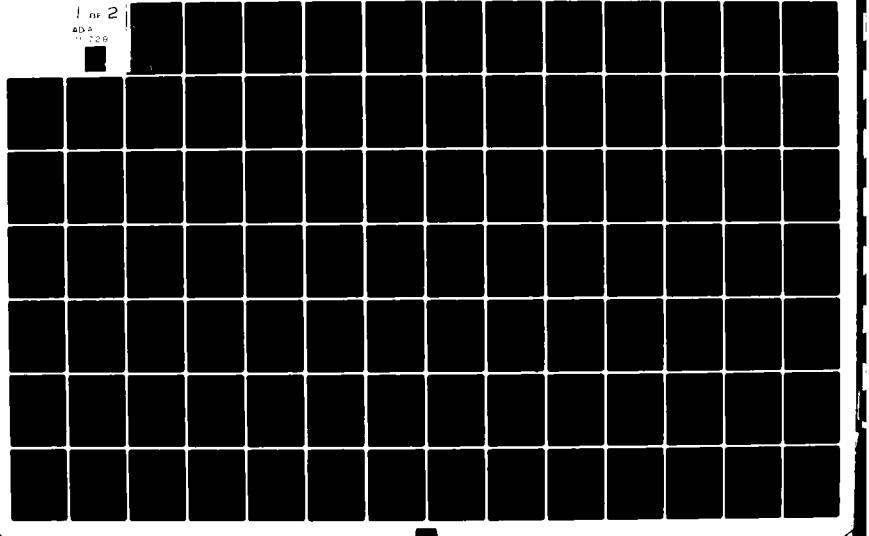
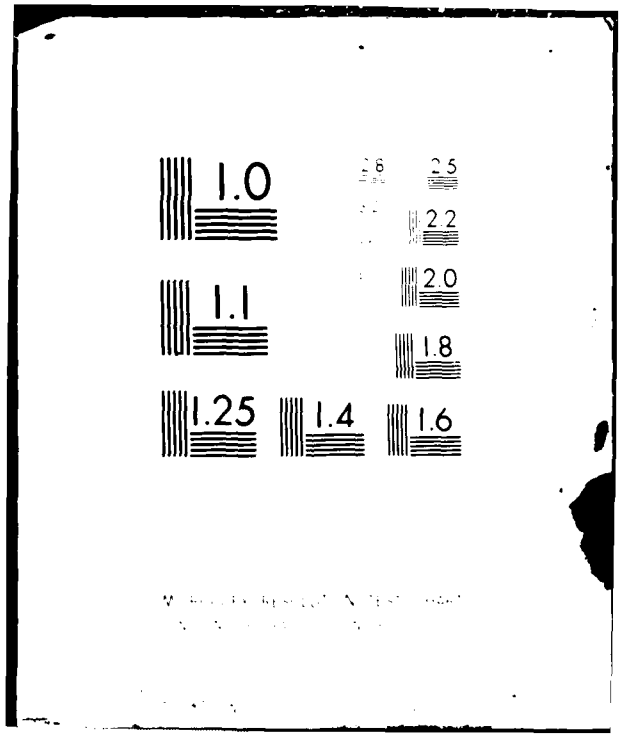
AD-A111 728 MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMA--ETC F/G 12/1
STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO--ETC(U)
JAN 82 F E BRUNEAU N00014-77-C-0532
UNCLASSIFIED LIDS-TH-1172 NL

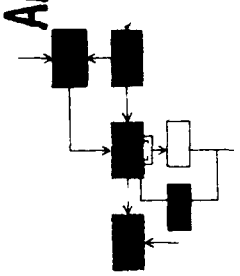



1.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

January, 1982

LIDS-TH-1172

Research Supported By:
Office of Naval Research
Contract N00014-77-C-0532
OSP Number 85552



STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO MULTITARGET TRACKING

Franck E. Brun au

Laboratory for Information and Decision Systems
MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER LIDS-TH-1172	2. GOVT ACCESSION NO. AD-A111 728	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* State Estimation of A Hybrid Markov Process with Application to Multitarget Tracking		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Franck E. Bruneau		8. CONTRACT OR GRANT NUMBER*(s)* N00014-77-C-0532
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information & Decision Systems Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Street, Code 411MA Arlington, VA 22217		12. REPORT DATE January 1982 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Failure detection; multitarget tracking; state estimation; Markov processes

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Many estimation problems, including failure detection and multitarget tracking, rely on the state estimation of a hybrid Markov process. An algorithm for estimating the state sequence of such a process must keep the amount of computation at a reasonable level.

Techniques are derived for reducing the computational burden, that do not increase the error probability.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

In a multiple system case, the concept of clustering, which greatly diminished the computational complexity, can be applied without loss of optimality under precise conditions. For the application to multitarget tracking, the computation of an optimal grating is done.

January, 1982 LIDS-TH-1172

STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO MULTITARGET TRACKING

by

Franck E. Bruneau

This report is based on the unaltered thesis of Franck E. Bruneau, submitted in partial fulfillment of the requirements for the degree of Master of Science at the Massachusetts Institute of Technology in January 1982. The research was conducted at the M.I.T. Laboratory for Information and Decision Systems with supported provided by the Office of Naval Research under ONR Contract N00014-77-C-0532.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO MULTITARGET TRACKING

by

FRANCK E. BRUNEAU

Diplôme d'Ingénieur de l'Ecole Supérieure d'Electricité
(1981)

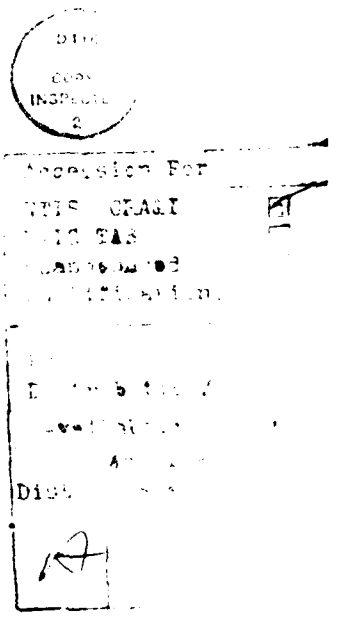
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1982



Signature of Author _____
Department of Electrical Engineering
and Computer Science, January 14, 1982

Certified by _____
Robert R. Tenney
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Departmental Graduate Committee

STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO MULTITARGET TRACKING

by

FRANCK E. BRUNEAU

Submitted to the Department of Electrical Engineering and Computer Science on January 14, 1982 in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering and Computer Science

ABSTRACT

Many estimation problems, including failure detection and multitarget tracking, rely on the state estimation of a hybrid Markov process. An algorithm for estimating the state sequence of such a process must keep the amount of computation at a reasonable level.

Techniques are derived for reducing the computational burden, that do not increase the error probability.

In a multiple system case, the concept of clustering, which greatly diminishes the computational complexity, can be applied without loss of optimality under precise conditions. For the application to multitarget tracking, the computation of an optimal gating is done.

Thesis Supervisor: Robert R. Tenney

Title: Assistant Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I would like to thank my advisor, Bob Tenney, for his guidance, encouragement and support during the course of this research. His insight and suggestions helped establish the overall direction of my research effort.

Thanks also must go to Walid Salman for many fruitful discussions.

I am grateful to the Ecole Supérieure d'Electricité for allowing me to complete my third year at MIT, and to the French Government for partially supporting me last year.

The research was conducted at the M.I.T. Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under ONR Contract N00014-77-C-0532.

TABLE OF CONTENTS

PART I

INTRODUCTION AND PROBLEM STATEMENT

CHAPTER 1

INTRODUCTION

Central to a broad class of problems is the state estimation of a hybrid Markov process, for which the underlying process consists of a Markov chain driving continuous dynamics. Examples wherein this issue arises include failure detection problems, where a system subject to failures can be considered as a single hybrid Gauss-Markov process, and multiobject tracking, where each of several targets can be modelled as a hybrid Gauss-Markov process.

The purpose of this work is to derive algorithms for estimating the state sequence of single and multiple hybrid Markov processes, using all available measurement data. A maximum a posteriori (MAP) decision criterion is used for the choice of the estimates, which leads to some nice recursive implementations. In the case of linear, Gaussian dynamics, these will take the form of coupled Viterbi and Kalman algorithms.

Since limited resources are available for the implementation of the algorithms, the amount of computation must be kept at a reasonable level: this suggests the use of some techniques for reducing the computational burden. While many ad hoc methods for doing this have been proposed (see [9]-[10]), this thesis derives techniques which achieve this goal with absolutely no increase in error probability.

For this purpose, we proceed step by step. Chapter Two will specify the mathematical model to be used, and Chapter Three the related work done in this area. In Part II, the single system problem

is discussed: Chapter 4 reviews the discrete state case only, Chapter 5 the continuous state case only, and the combination for a hybrid state estimation is developed in Chapters 6-7. Then, Part III studies the multiple hybrid system case. The main issue in this part is to obtain a partial decomposition of our estimation problem into several independent subproblems: this is the notion of clustering, which greatly reduces the computational complexity. An application to multitarget tracking is discussed in Chapter 11, where we prove the existence of an optimal gating leading to the decomposition of a multitarget tracking problem into independent single target tracking problems.

CHAPTER 2

FORMAL PROBLEM STATEMENT

In this chapter, we formulate the MAP estimation problems for the single and multiple hybrid Markov processes in relation to the two basic examples that motivate them: failure detection and multitarget tracking.

Since the purpose of our work is to derive computational algorithms all events are assumed to occur at discrete instants of time $k = 0,1,2...$ Throughout this work, the use of the superscript $(\ )^k$ will denote the sequence up to and including k: $a^k = (a(0),...,a(k))$.

2.1 Necessity for a Hybrid State Description

In many estimation problems, the description of the state of a dynamic system only in terms of a continuous state is not sufficient.

For instance, in failure detection problems, the system considered is subject to abrupt changes (failures or repairs). The normal operation, or no failure, model of the system is described in standard state space form:

system dynamics: $x(k+1) = F\,x(k) + G\,u(k) + w(k)$

sensor equation: $z(k) = H\,x(k) + v(k)$

The abrupt changes can arise in a number of ways; they can manifest themselves as actuator failures: shift in the control gain matrix G or increased process noise; sensor failures: abrupt changes in H or increase in measurement noise; or as actual changes in the plant dynamics F. The mode (failed or unfailed) under which the system operates at a given

time is a discrete state complementing the usual continuous state, whose dynamics can be described by a state-transition diagram (see Fig. 2.1). This combined hybrid state provides the most complete description of the system.

Similarly, in multitarget tracking problems, it seems natural to describe the set of targets with a hybrid process. In an area under surveillance, an unknown number of targets are moving; some new targets can arrive suddenly, while others can disappear. This process can be simply characterized using an augmented state for each target. We assume a large but fixed number of targets in the area, many of which are dead or in discrete state 0. A new arrival is the transition for a target between state 0 and the alive discrete state 1; a disappearance is the reverse transition. Thus, each target is described with a hybrid state: discrete state 0 or 1 and usual continuous states; e.g., positions and velocities. With more complex discrete state models, a variety of phenomena such as maneuvers, signal fading, and variable configurations can be modeled (see Fig. 2.2).

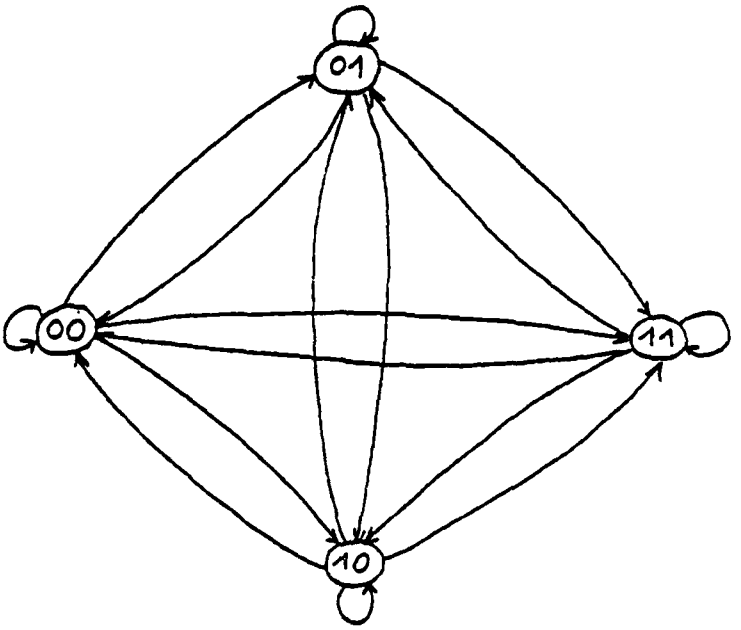
In both examples, we see that the introduction of a hybrid state description for the system provides the most compact and complete characterization of the process.

We next formulate the single hybrid state estimation problem.

2.2 The Single Hybrid State Estimation Problem

In this section, we develop the single hybrid Markov process model and formulate our objective.

FIGURE 2.1



Two types of failures : 4 discrete states

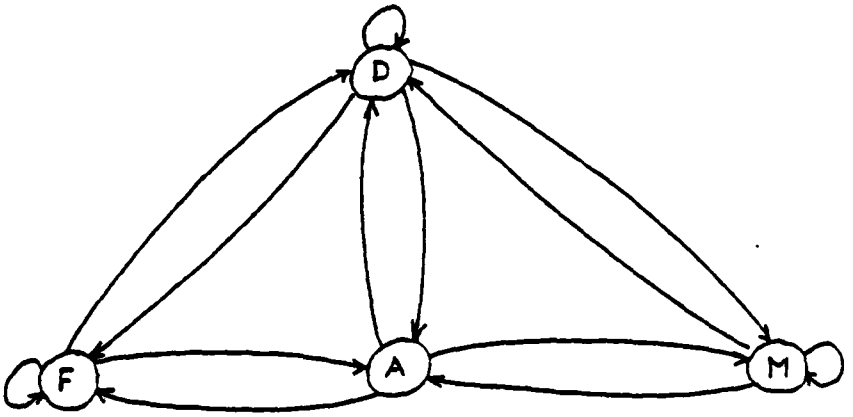
00 : actuator and sensor unfailed

11 : actuator and sensor failed

01 : actuator failed, sensor unfailed

10 : sensor failed, actuator unfailed.

FIGURE 2.2



D : dead state

A : alive state

F : fading state

M : maneuver state

The state s(k) of a single hybrid Markov process at time k belongs to a state space S. It consists of both a discrete state d(k) and a continuous state x(k). The discrete state d(k) belongs to a finite set $D = \{1,\ldots,N\}$. The continuous state x(k) belongs to a state space X of finite dimension n.

The dynamics of s(k) are assumed to follow a Markov process, in the sense that the probability distribution $P(s(k+1)|s(k),\ldots,s(0))$ of being in state s(k+1) at time k+1, given all states up to time k, depends only on the state s(k) at time k:

$$P(s(k+1)|s^k) = P(s(k+1)|s(k)) \tag{2.1}$$

Furthermore, the discrete state d(k+1) is assumed to depend only on the discrete state d(k) at time k. So, $P(s(k+1)|s(k))$ can be written as:

$$P(s(k+1)|s(k)) = P(x(k+1)|x(k),d(k),d(k+1)) \times P(d(k+1)|d(k)) \tag{2.2}$$

The system is observed during an interval [0,K], and, for simplicity, we assume that the measurement process is continuous, although all the results can be generalized to a hybrid measurement process at the cost of more cumbersome notation.

At each discrete instant of time k, a continuous measurement z(k) of the hybrid state s(k) is generated. The observation z(k) belongs to a state space Z of finite dimension m. The measurement process is assumed to be memoryless: the probability distribution for the measurement z(k) depends only on the state of the system at time k:

$$P(z(k)|s^k,z^{k-1}) = P(z(k)|s(k)) \tag{2.3}$$

A further assumption often made on dynamic systems, when the system equations can be linearized and a good approximation of the physical noises are given by Gaussian densities in the Linear-Gaussian assumption. Under this assumption, the distributions $P(x(k+1)|x(k), d(k) = p, d(k+1) = q)$ and $P(z(k)|x(k), d(k) = p)$ are defined by the following relations:

$$x(k+1) = F(p,q,k)x(k) + w(p,q,k) \tag{2.4}$$

$$z(k) = H(p,k)x(k) + v(p,k) \tag{2.5}$$

$F(p,q,k)$, $H(p,k)$ are known $n \times n$ and $m \times n$ time-varying matrices, that depend also on the discrete state of the system. $w(p,q,k)$, $v(p,k)$, $x[d(0)]$ are independent zero-mean White Gaussian processes with:

$$\begin{aligned} E\{w(p,q,k)w^T(p,q,k)\} &= Q(p,q,k) > 0 \\ E\{v(p,q,k)v^T(p,q,k)\} &= R(p,q,k) > 0 \\ E\{x[d(0)]x^T[d(0)]\} &= P(d(0)) > 0 \end{aligned} \tag{2.6}$$

In a failure detection application, the equations under the normal mode ($d(k) = 0$) are:

$$x(k+1) = F\,x(k) + w(k)$$

$$z(k) = H\,x(k) + v_0(k)$$

Then, if a sensor failure arises at time k ($d(k) = 1$), the measurement equation becomes:

$$z(k) = v_1(k)$$

Therefore: $H(0,k) = H$, $v(0,k) = v_0(k)$, $H(1,k) = 0$, $v(1,k) = v_1(k)$.

The model developed above is the basis for Part II of our work, which discusses the problem of estimating the state of a single hybrid Markov process. There can be two possible objectives in this estimation, depending on the application.

For instance, in failure detection problems, if the system has sufficient hardware back-up capabilities, then the objective of the failure detection system is to quickly detect the failures: this is done by estimating the discrete state sequence of the system. But, if there is not enough redundancy in the system, then the system has to operate under the degraded mode. In this case, the objective of our failure detection system is, at the same time, to detect the failures and to perform an estimation of the continuous state.

So, one possible objective is the MAP estimation of the discrete state sequence d^K, based on z^K, that minimizes the probability of an error in estimating d^K. The problem is to find $\hat{d}^K$ such that:

$$P(\hat{d}^K | z^K) \geq P(d^K | z^K), \quad \text{for all } d^K \in D^K \tag{2.7}$$

In other cases, where we are interested in the estimation of the hybrid state sequence s^K, the objective is the MAP estimation of s^K, based on z^K. The problem is to find $\hat{s}^K$ such that:

$$P(\hat{s}^K | z^K) \geq P(s^K | z^K), \text{ for all } s^K \in S^K \tag{2.8}$$

In both cases, an algorithm for estimating the state sequence (discrete or hybrid) of the single hybrid system must keep the amount of computation at a reasonable level, since limited computational resources are available.

We next consider the more complex multiple hybrid state estimation problem.

2.3 The Multiple Hybrid State Estimation Problem

In this section, we present the multiple hybrid Markov process model, formulate our objective, and see how it relates to the multitarget tracking problem.

The multiple hybrid system consists of a finite collection of single hybrid Markov processes. The state $s(k)$ at time k is equal to:

$$s(k) = \{s_1(k), \ldots, s_p(k)\} \text{ where for all } i = 1,\ldots,p,$$

the state $s_i(k) = \{d_i(k), x_i(k)\}$ of subsystem i belongs to S.

The subsystems are assumed to have independent dynamics. Therefore, the probability distribution $P(s(k+1)|s(k))$ is equal to:

$$P(s(k+1)|s(k)) = \prod_{i=1}^{p} P(s_i(k+1)|s_i(k)) \tag{2.9}$$

and

$$P(s_i(k+1)|s_i(k)) = P(x_i(k+1)|d_i(k),d_i(k+1),x_i(k)) \times$$

$$P(d_i(k+1)|d_i(k)) \tag{2.10}$$

As mentioned in Section 1, in multitarget tracking, the discrete state of a target $d_i(k)$ may take the value 1 if the target is present in the area at time k, and 0 if it is absent. The continuous state $x_i(k)$ consists of the positions and velocities of target i. The independent dynamics assumption means that we consider independent positions and velocities of targets and independent arrivals or disappearances.

The independent arrival assumption is certainly the most difficult to accept, since an arrival may be less likely to occur, if there are already 500 targets in the area, rather than only 5 targets.

The continuous measurement generation process for the multiple hybrid system is the following: each subsystem i generates a set of measurements $Y_i(k)$ at time k: if $Y_i(k) = \{z_{i,1}(k),\ldots,z_{i,q}(k)\}$, then for all $j = 1,\ldots,q$, $z_{i,j}(k)$ belongs to Z. We can have multiple detections corresponding to the same subsystem in the case of multiple, separate sensors. We allow $Y_i(k)$ to be the empty set, in which case we say that subsystem i has not been <u>detected</u> at time k. Besides, because of the noisy environment, some measurement data may appear independent from <u>all</u> of the system states: this set of measurements is the set of <u>false alarms</u> and is denoted by $Y_f(k)$.

In a multitarget environment, a missed detection can arise because of a sensor failure, or some natural obstacle between a sensor and the target. A false alarm could come, for instance, from a flock of geese, when the surveillance system is observing aircrafts.

Each subsystem is assumed to generate its set of measurements independently of the other subsystems and of the false alarms. This is a reasonable assumption in multitarget tracking. Therefore:

$$P(Y_1(k),\ldots,Y_p(k),\ Y_f(k)\,|\,s(k)) = \prod_{i=1}^{p} P(Y_i(k)\,|\,s_i(k))\times P(Y_f(k)) \tag{2.11}$$

where for all $i = 1,2,\ldots,p$, $P(Y_i(k)\,|\,s_i(k))$ and $P(Y_f(k))$ are assumed to be known.

As for the single system case, we have an additional Linear-Gaussian assumption. For simplicity, we assume that only one sensor is used. In

this case, each subsystem i generates at most one measurement $z_i(k)$ at time k. For each subsystem $i = 1,\ldots,p$, $P(x_i(k+1)|x_i(k), d_i(k) = q, d_i(k+1) = r)$ and $P(z_i(k)|x_i(k), d_i(k) = q)$ are defined as for the single system model by:

$$x_i(k+1) = F_i(q,r,k)x_i(k) + w_i(k) \tag{2.12}$$

$$z_i(k) = H_i(q,k)x_i(k) + v_i(k) \tag{2.13}$$

with

$$\begin{aligned} E\{w_i(q,r,k)w_i^T(q,r,k)\} &= Q_i(q,r,k) \\ E\{v_i(q,r,k)v_i^T(q,r,k)\} &= R_i(q,r,k) \\ E\{x_i[d(0)]x_i^T[d(0)]\} &= P_i(d(0)) \end{aligned} \tag{2.14}$$

The false alarm process is such that each false alarm $z_f(k)$ is generated independently of the other false alarms; we will see in part III, why we do not assume a Poisson distribution for the number of false alarms. The probability distribution for a false alarm $z_f(k)$ is a zero-mean Gaussian density with a known covariance Σ.

For the multitarget tracking application, if we assume planar straight-line motions and position measurements, then we have:

$$x_i(k) = \begin{bmatrix} x_i^1(k) \\ v_i^1(k) \\ x_i^2(k) \\ v_i^2(k) \end{bmatrix} \text{ in the plane } (x^1,x^2), \text{ and}$$

$$x_i(k+1) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i(k) + w_i(k) \quad ,$$

$$z_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_i(k) + v_i(k) \quad .$$

The problem of estimating the discrete or hybrid state sequence of the multiple system is much more complex than the simple juxtaposition of p estimation problems, one for each subsystem.

In the multitarget case, the objective of the surveillance system is track the existing targets, to announce the new targets and the disappearances of old targets. But, because of the cluttered environment the tracker does not know which measurement originated from a given target and which ones are the false alarms. Therefore, there exists an uncertainty in the origin of the measurements; and any target state estimation has to take into account this uncertainty.

For the general estimation problem, the actual set of observations is denoted by $Y(k) = \{z_1(k),\ldots,z_q(k)\}$. The uncertainty is introduced by a physical permutation P(k), which is a map from the multiple system and the set of false alarms onto the elements of Y(k). This permutation is unknown; therefore, a hypothesis on it is made by the estimator, that is called a data hypothesis at time k and is denoted by A(k). Each data hypothesis A(k) has the following features:

- A(k) is a mapping from the multiple system s(k) into the set of observations Y(k),

- Each subsystem i is associated with zero, one, or more measurements of Y(k). A subsystem associated with no measurement is naturally called a missed detection under A(k).

- Each measurement of Y(k) can come from zero, one, or more subsystems of s(k). A measurement associated with no subsystem is considered as a false alarm under A(k). We illustrate this mapping in Fig. 2.3.

We assume that all the associations of a measurement to a subsystem are a priori likely. Therefore, the probability $P(A(k)|s(k))$ of a data hypothesis A(k) at time k, depends only on the numbers of detections, missed detections, and false alarms under A(k). Furthermore, we assume for simplicity that $P(A(k)|s(k))$ depends only on the discrete state d(k), i.e. $P(A(k)|s(k)) = P(A(k)|d(k))$. In multitarget tracking, this means that the probability of detecting a target, and the probability of a false alarm are uniform in the area. If we make this assumption, then $P(A(k)|d(k))$ takes the form:

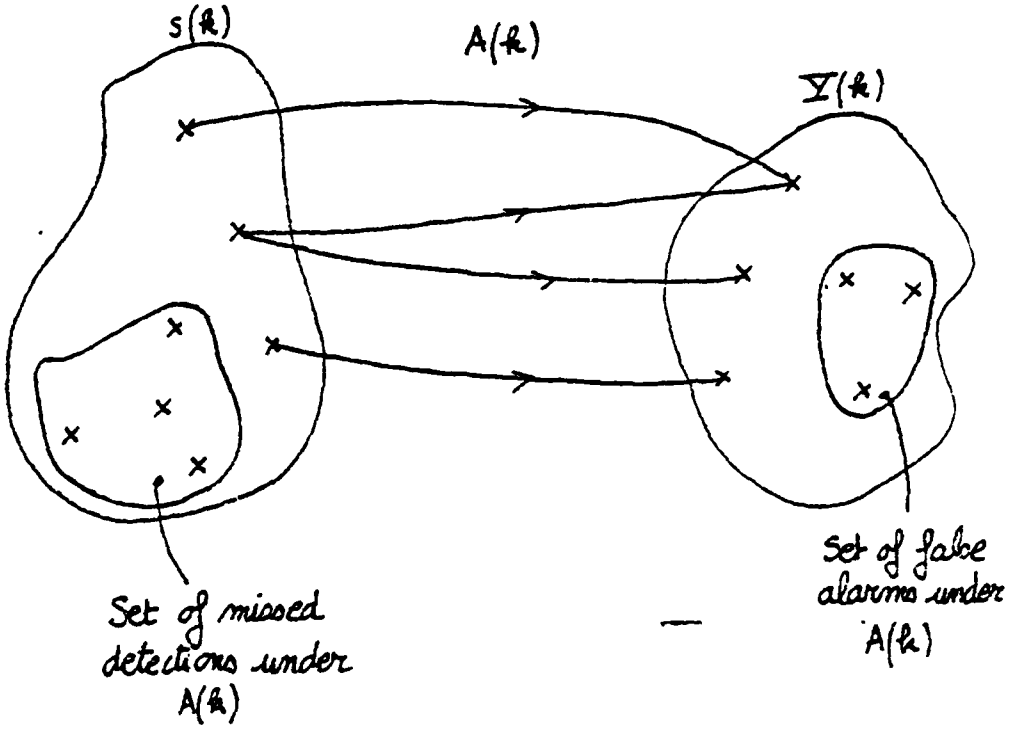
$$P(A(k)|d(k)) = \Pr\{M_d \text{ detections under } A(k)\} \times \Pr\{M_m \text{ missed detections under } A(k)\} \times \Pr\{M_f \text{ false alarms under } A(k)\}.$$

This data association uncertainty complicates the estimation problems as follows.

As for the single system model, there are two different objectives in a multiple hybrid state estimation problem.

The first possible objective is the MAP estimation of the hybrid state sequence s^K, based on Y^K. The problem is to find $\hat{s}^K \in (S^p)^K$ for which $P(s^K|Y^K)$, or equivalently, $\prod_{k=0}^{K} P(Y(k)|s(k)) \times P(s(k)|s(k-1))$ is

FIGURE 2.3



DATA HYPOTHESIS AT TIME k

maximum. But now, $P(Y(k)|s(k))$ must be evaluated from:

$$P(Y(k)|s(k)) = \sum_{\{A(k)\}} P(Y(k)|s(k), A(k)) \times P(A(k)|d(k)), \quad (2.15)$$

and

$$P(Y(k)|s(k),A(k)) = \prod_{i=1}^{p} P(Y_i(A(k))|s_i(k)) \times P(Y_f(A(k))), \quad (2.16)$$

where the summation is over all possible data hypotheses, and $\{Y_1(A(k)),\ldots,Y_p(A(k)),Y_f(A(k))\}$ is the hypothesized partition of $Y(k)$ under $A(k)$.

There exists also a second possible objective. For instance, in some multitarget tracking problems, we can be interested in first deciding upon the presence of targets in the area, then in locating them based on the above decisions. In this case, the objective is rather the MAP estimation of d^K based on Y^K, which minimizes the probability of an error in estimating d^K. The problem is to find $\hat{d}^K \in (D^p)^K$ for which $P(d^K|Y^K)$, or equivalently, $\prod_{k=0}^{K} P(Y(k)|d^k,Y^{k-1}) \times P(d(k)|d(k-1))$ is maximum. $P(Y(k)|d^k,Y^{k-1})$ must be evaluated from:

$$P(Y(k)|d^k,Y^{k-1}) = \sum_{\{A^k\}} P(Y(k)|d^k,Y^{k-1},A^k) \times P(A^k|d^k) \quad (2.17)$$

and

$$P(Y(k)|d^k,Y^{k-1},A^k) = \prod_{i=1}^{p} P(Y_i(A(k))|d_i^k,Y_i^{k-1}(A^{k-1})) \times$$

$$P(Y_f(A(k)) \quad . \quad (2.18)$$

Under both objectives, we must find an algorithm for estimating the state of the multiple system, that keeps the amount of computation at a reasonable level, with respect to our computational resources.

2.4 Summary

In this chapter, we have developed the basic models and objectives for the single and multiple hybrid Markov processes, in relation to the failure detection and multitarget tracking problems.

We next review some previous work done in these areas, and compare it with the approach taken here.

CHAPTER 3

BACKGROUND AND LITERATURE SURVEY

In this Chapter, we compare the formulation and objectives of the last chapter, with some previous work done in failure detection and multitarget tracking.

3.1 Failure Detection

A survey of design methods for failure detection systems has been done by Willsky in [5].

One of these methods is closely related to our approach: this is the multiple hypothesis filter-detectors, in which a bank of Kalman filters is constructed, based on the different hypotheses concerning the system behavior. This method computes the probability of each type of failure under consideration. The problem is the exponentially growing bank of filters, so that some ad-hoc pruning techniques have to be used, to limit the computational complexity: unlikely hypothesis sequences are simply dropped during their formation, or a single shift in the system behavior is assumed every K steps.

The problem of detecting a single switch between two dynamic models (failed, unfailed) is also treated in [6]-[7], using a sequential probability ratio test (SPRT). The SPRT optimizes a combination of the time to reach a decision and the probabilities of making wrong decisions.

Thus many "optimal" algorithms for detecting failures use some ad-hoc procedures in order to meet some reasonable time and storage require-

ments, at the expense of a degradation in the performance.

We have seen in Chapter 2 that the single hybrid state estimation problem applies to the failure detection problem. In particular, a hypothesis in the multiple hypothesis filter-detectors corresponds to a discrete state sequence in our formulation. The objective of part II is to find an algorithm, that uses truly optimal pruning techniques to reduce the necessary amount of computation. If this algorithm meets also the storage requirements, then we would get an optimal algorithm for the failure detection problem.

3.2 Multitarget Tracking

An excellent survey of recent work in multitarget tracking methods (up to 1978) has been done by Bar-Shalom [8].

An important paper in this field has been given by Reid [9]. A set of data-association hypotheses that take into account all possible origins of every measurements (prior targets, new targets, and false alarms) is generated and organized sequentially into a tree. For each hypothesis, a set of Kalman Filters is constructed to track the targets as implied by that hypothesis. The probability of each hypothesis is computed sequentially, using the predicted value of the measurement obtained from the state estimates of the filters: the closer a measurement is to a particular prediction, the higher the probability that the measurement is associted with the corresponding hypothesis. A key idea to reduce computation is clustering. In the hypothesis tree, any branch whose probability lies below a threshold is dropped. The probability of a

hypothesis below the threshold is equivalent to the present measurement falling outside a gate drawn around the prediction. If the gates of the various data hypotheses do not overlap, then the set of received measurements can be partitioned into several subsets, each subset containing the measurements falling outside a gate; and the hypothesis tree can be broken up into smaller, independent trees. If some of the gates do overlap, then a tree can be formed for the cluster of the corresponding data hypotheses. This clustering leads to a great simplification, since the total number of hypotheses in the smaller trees will be much less than the number of hypotheses in the global tree. A new target is hypothesized, as each new measurement is received; it is confirmed, if after dropping the unlikely hypotheses, the target has a probability of existing equal to one. The possibility of a target ceasing to exist is not covered by the paper.

Thus, this algorithm uses some suboptimal pruning techniques, in order to reduce the amount of computation, and some ad-hoc procedures to decide upon the arrival of a new target.

Keverian modifies slightly this algorithm in [10] and Hughes [11] applies it to a distributed system.

An early work by Sittler [12] proposed to compute sequentially a score function, using the inverse of a likelihood function; a new track is initiated when the score function stays above a threshold, and is doppred when it goes below another one. A modernized version of this algorithm has been done by Stein and Blackman in [13].

Morefield [14] proposed a solution for the most likely data association in an integer-programming framework; it is a batch processing technique.

Alspach [15] computes the a posteriori density of the state of the target, which is a Gaussian Sum density because of the different possible data hypotheses to this target. After dropping the terms in the sum with small weighting coefficients, the state estimate is found as a convex combination of the mean of the individual terms.

Another approach is taken by Bar-Shalom [16]-[17]: given a target state has been initialized, the probability that a current measurement comes from that target is computed. Then all the measurements, weighted by these probabilities, are used to update the target state estimate. This is the Probabilistic Data Association Filter. No initialization procedure is included in this algorithm.

We have seen in Chapter 2 that the multiple hybrid state estimation problem applies to the multitarget tracking problem. In our formulation, we take a target-oriented approach: each target has an a priori probability of appearing and disappearing, then the received measurements will confirm or deny its presence in the area. Contrary to Reid's algorithm, this allows a systematic procedure for track initiations and terminations, since the MAP hybrid state estimation algorithm decides upon the discrete state of each target. To keep the amount of computation at a reasonable level, this algorithm must use some pruning techniques in the tree of the possible discrete state sequences of the multiple system. We are interested only in pruning rules which never degrade performance, not in some ad-hoc heuristics; this applies to part II as well as part III.

Thus, the purpose of our work in part III is to find some optimal pruning rules during the formation of this tree, and particularly to

examine the existence of optimal gating, that could greatly reduce the computational complexity, without increasing the probability of an error in the estimation.

The resulting algorithm for estimating the multiple state sequence, even if it exceeds the computational requirements using these rules and gating would still provide a basis for rational (if not optimal) procedures for deciding upon new arrivals and disappearances in the area, and for locating the targets.

3.3 Summary

In this chapter, we have reviewed some previous work done in failure detection and multitarget tracking, and we have compared it with our formulation and objectives.

We next consider our first problem: the single hybrid state estimation.

PART II

THE SINGLE SYSTEM CASE

CHAPTER 4

STATE ESTIMATION OF A MARKOV CHAIN: THE VITERBI ALGORITHM

In this chapter, we describe the Viterbi Algorithm [1], which is a recursive, optimal solution to the problem of estimating the state sequence of a finite Markov chain observed in memoryless noise. This algorithm has many applications to digital estimation problems: convolutional coding, where an encoded sequence is sent through a memoryless channel; digital transmission through analog channel.

4.1 Problem Statement

In this part, a single hybrid state system is considered. The system model is the one developed in Chapter 2, Section 2. Here, the continuous state space X reduces to 0, so that only the discrete state part is considered.

The Markov and memoryless properties (2.1)-(2.3) become:

$$P(d(k+1)|d^k) = P(d(k+1)|d(k)) \tag{4.1}$$

$$P(z(k)|d^k, z^{k-1}) = P(z(k)|d(k)) \tag{4.2}$$

These probabilities may be time-varying, but we do not explicitly indicate this in the notation.

The system is observed during an interval [0,K].

The objective is to minimize the probability of an error in estimating the whole sequence $d^K = (d(0),\ldots,d(K))$. The solution to this problem is given by the state sequence $\hat{d}^K$ for which the a posteriori probability $P(d^K|z^K)$ is maximum. In short, the problem is to find $\hat{d}^K$

such that:

$$\forall d^K \in D^K,\ P(\hat{d}^K | z^K) \geq P(d^K | z^K) \quad .$$

The straightforward computation of $\hat{d}^K$ is combinatorial: find the most likely state sequence among N^K state sequences.

Using the Markov and memoryless properties (4.1)-(4.2), we can show that this estimation problem is formally identical to the problem of finding the shortest path through a certain graph, which requires much less computation. This is the basic idea in the Viterbi Algorithm.

4.2 The Algorithm

A graph can be associated with our Markov chain. Each node corresponds to a distinct state at a given time and each branch represents a transition to some new state at the next instant of time. The graph begins at time 0 with a state d_0 that is assumed to be known. A trivial extension can be made to the case where d(0) is given by some a priori probability distribution function. To every possible discrete state sequence d^K, there corresponds a unique path through the graph, and vice-versa.

The problem of finding d^K for which $P(d^K | z^K)$ is maximum, or equivalently for which $P(d^K, z^K) = P(d^K | z^K) \times P(z^K)$ is maximum, is the same as the one of finding the path in the graph whose length $\lambda(d^K) = -\ell n\ P(d^K, z^K)$ is minimum.

Due to the Markov and memoryless properties (4.1)-(4.2), we have:

$$P(d^K, z^K) = \prod_{k=0}^{K} P(d(k) | d(k-1)) \times P(z(k) | d(k)) \quad .$$

If we assign each branch the length $\lambda(d(k), d(k-1)) \triangleq -\ell n\ P(d(k) | d(k-1))$

$-\ln P(z(k)|d(k))$, that depends on the received measurement $z(k)$ at time k, then:

$$\lambda(d^K) = \sum_{k=0}^{K} \lambda(d(k),d(k-1)) \tag{4.3}$$

the total length of a path is equal to the sum of the length of the branches composing the path.

In the graph, d^k corresponds to a path starting at the node d_0 and terminating at $d(k)$. For any particular node $d(k)$, there will be several such paths. The shortest path from d_0 to $d(k)$ is called the survivor corresponding to $d(k)$ and is denoted $\hat{d}[d(k)]$. For any time $k > 0$, there are N survivors in all, one for each $d(k)$.

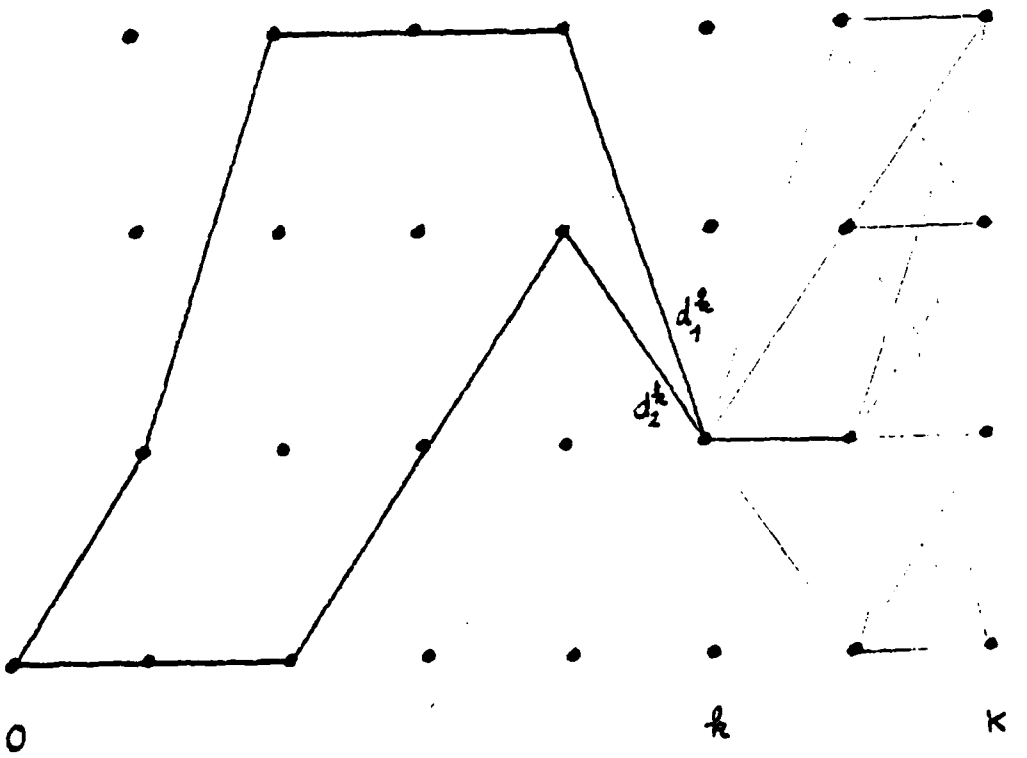
The shortest complete path $\hat{d}^K$ must begin with one of these survivors. This is true because of the following property: let d_1^k and d_2^k be two state sequences up to time k, such that $d_1(k) = d_2(k)$. If $\lambda(d_1^k) \leq \lambda(d_2^k)$, then, for all $z(k+1),\ldots,z(K)$, for all $d(k+1),\ldots,d(K)$, $\lambda(d_1^K) \leq \lambda(d_2^K)$, where $d_1^K = (d_1^k,d(k+1),\ldots,d(K))$ and $d_2^K = (d_2^k,d(k+1),\ldots,d(K))$, because of property (4.3). We illustrate this property in Figure (4.1).

Therefore, d_2^k cannot be a part of the shortest path $\hat{d}^K$ and can be immediately discarded at time k.

Thus, only the N survivors $\hat{d}[d(k)]$ and their length $\Gamma[d(k)] \triangleq \lambda[\hat{d}(d(k))]$ need to be stored at time k.

The recursion proceeds as follows: each survivor $\hat{d}[d(k)]$ is extended by one time unit and the length of the extended paths are computed, using the received measurement at time $k+1$. Then, for each node $d(k+1)$, the shortest path terminating at $d(k+1)$ is selected as the cor-

FIGURE 4.1



responding survivor $\hat{d}[d(k+1)]$. This procedure is repeated until $k = K$ and the number of survivors never exceeds N. Thus, at time K, we have N survivors $\hat{d}[d(K)]$. The shortest of these survivors corresponds to $\hat{d}^K$ the most likely discrete state sequence.

The algorithm can be stated formally as follows:

INITIALIZATION: $k = 0$

$$\hat{d}[d(0)] = d_0$$

$$\Gamma[d(0)] = 0$$

STEP k: $\hat{d}[d(k)] \quad 0 \leq d(k) \leq N$

$\Gamma[d(k)] \quad 0 \leq d(k) \leq N$

Compute $\Gamma[d(k+1), d(k)] \triangleq \Gamma[d(k)] + \lambda[d(k+1), d(k)]$

for all $d(k+1), d(k)$.

Find $\Gamma[d(k+1)] = \min_{d(k) \in D} \Gamma[d(k+1), d(k)]$

Store $\Gamma[d(k+1)]$ and the corresponding $\hat{d}[d(k+1)]$

$k = k+1$ and repeat until $k = K$.

STEP K: $\hat{d}[d(K)] \quad 0 \leq d(K) \leq N$

$\Gamma[d(K)] \quad 0 \leq d(K) \leq N$

Find $\min_{d(K) \in D} \Gamma[d(K)]$.

The corresponding survivor $\hat{d}[d(K)]$ is $\hat{d}^K$. END.

The main feature in this algorithm is that we need only to store N paths at each time $k = 0, 1, \ldots, K$. A straightforward computation of $\hat{d}^K$ would have needed the storage of N^K paths at time K.

In practice, certain trivial modifications are necessary. This is the subject of the following section.

4.3 Actual Implementation

When the state sequences are very long or infinite, it is necessary to truncate the survivors to some manageable length K_r. The algorithm must decide on nodes up to time $k - K_r$ at time k. If the truncation depth K_r is chosen large enough, there is a high probability that all time-k survivors will go through the same nodes up to time $k - K_r$; in this case, the truncation costs nothing. If not, then we choose the time $(k - K_r)$ node associated with the shortest survivor. If K_r is large enough, then the effect on the performance is negligible.

4.4 Summary

In this chapter, we have presented the Viterbi algorithm, which is an optimal solution to the problem of estimating the state sequence of a discrete Markov process.

We next consider its continuous counterpart, the MAP state estimation of a continuous Markov process.

CHAPTER 5

STATE ESTIMATION OF A CONTINUOUS MARKOV PROCESS - KALMAN-BUCY EQUATIONS

In this chapter, we discuss the problem of estimating the state of a continuous Markov process. The Bayesian formulation leads to a recursive form for computing the MAP estimates. We will see that the concepts are identical to the discrete state. An additional Linear-Gaussian assumption gives some computational elegance with the Kalman Equations.

5.1 Problem Statement

The system considered is a continuous dynamic system observed during an interval [0,K]. The system model is the one developed in Chapter 2 Section 2. Here, at each time k = 0,1,...,K, the discrete state d(k) can only take one value, so that the dependence in d(k) is dropped. Thus, we have:

$$P(x(k)|x^{k-1}) = P(x(k)|x(k-1)) \tag{5.1}$$

$$P(z(k)|x^{k},z^{k-1}) = P(z(k)|x(k)) \tag{5.2}$$

We have also the following simplified equations for the Linear Gaussian assumption:

$$x(k+1) = F(k)x(k) + w(k) \tag{5.3}$$

$$z(k) = H(k)x(k) + v(k) \quad , \tag{5.4}$$

where $x(0)$, $w(k)$, $v(k)$ are independent zero-mean White Gaussian Processes with:

$$\begin{aligned} E\{x(0)x^T(0)\} &= P_0 \\ E\{w(k)w^T(k)\} &= Q(k) \\ E\{v(k)v^T(k)\} &= R(k) \quad . \end{aligned} \tag{5.5}$$

In many estimation problems, the objective is to find, at each time $k = 0,1,\ldots,K$, an MAP estimate of the instantaneous state $x(k)$ of a system, based on the available measurement data at that time: this is a filtering problem.

However, in some cases, a refinement of this estimation is necessary. For instance, in the launch of a satellite, it is desirable to determine the performance of the guidance and propulsion systems during the initial phases of the flight, so that an estimation of the state trajectory has to be done, using measurement data beyond the current time. This is a smoothing problem, whose objective is the MAP estimation of the state sequence $x^K = (x(0),\ldots,x(K))$, based on all available measurement data.

5.2 The Filtering Problem

As mentioned above, the objective in the filtering problem is to find for each time $k = 0,1,\ldots,K$, the MAP estimate of the state $x(k)$, which is given by $\hat{x}(k|k)$ for which the a posteriori density $P(x(k)|z^k)$ is maximized. In short, the problem is: for all $k = 0,1,\ldots,K$, find $\hat{x}(k|k)$ such that:

$$\forall\, x(k) \in X,\ P[\hat{x}(k|k)|z^k] \geq P[x(k)|z^k] \quad .$$

This estimation problem consists first in the determination of the

a posteriori density $P(x(k)|z^k)$, then in the maximization of this function over the state space X.

Using Bayes' rule and (5.1) and (5.2), we can determine recursively the density $P(x(k)|z^k)$ from the following relations:

for all $k = 0,1,\ldots,K$,

$$\text{update:} \quad P(x(k)|z^k) = \frac{P(z(k)|x(k))}{P(z(k)|z^{k-1})} \times P(x(k)|z^{k-1}) \tag{5.6}$$

$$\text{normalize:} \quad P(z(k)|z^{k-1}) = \int_X P(x(k)|z^{k-1}) \times P(z(k)|x(k))dx(k) \tag{5.7}$$

$$\text{predict:} \quad P(x(k)|z^{k-1}) = \int_X P(x(k)|x(k-1)) \times P(x(k-1)|z^{k-1})dx(k-1) \tag{5.8}$$

$$\text{initialize:} \quad P(x(0)|z^{-1}) \triangleq P(x(0)) \quad .$$

Then, having computed $P(x(k)|z^k)$, we use standard optimization techniques to determine the state estimate $\hat{x}(k|k)$.

The main difficulty in this problem lies in the explicit determination of $P(x(k)|z^k)$, so that it is often impossible to determine the MAP estimate $\hat{x}(k|k)$.

However, under the additional Linear-Gaussian assumption (5.3)-(5.5), the density $P(x(k)|z^k)$ is Gaussian, and so is completely defined in terms of a finite set of parameters: the conditional expectation $E\{x(k)|z^k\}$ and the covariance $E\{(x(k) - E\{x(k)|z^k\})(x(k) - E\{x(k)|z^k\})^T|z^k\}$. This introduces a great simplification in our estimation problem, as we shall see in the next section with the Kalman Filtering Equations.

5.3 The Kalman Filtering Equations

Since $P(x(k)|z^k)$ is a Gaussian distribution, the value $\hat{x}(k|k)$ for which $P(x(k)|z^k)$ is maximum is simply the conditional mean $E\{x(k)|z^k\}$.

In 1960, Kalman [2] derived a method for computing recursively $\hat{x}(k|k)$ and the corresponding covariance $P(k|k)$.

Throughout this work, $\hat{x}(i|j)$ denotes the optimal estimate of $x(i)$ based on measurement data up to time j, and $P(i|j)$ the corresponding covariance matrix.

The Kalman Filtering Equations compute recursively $\hat{x}(k|k)$ and $P(k|k)$ as follows:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)[z(k) - H(k)\hat{x}(k|k-1)] \tag{5.10}$$

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) \tag{5.11}$$

$$P(k|k) = [I - K(k)H(k)]P(k|k-1) \tag{5.12}$$

$$P(k+1|k) = F(k)P(k|k)F^T(k) + Q(k) \tag{5.13}$$

$$K(k) = P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1} \tag{5.14}$$

$$\hat{x}(0|-1) = x_0 \qquad P(0|-1) = P_0 \tag{5.15}$$

$F(k)$, $H(k)$, $Q(k)$, $R(k)$, P_0 are defined by (5.3)-(5.5).

Thus, the a posteriori density $P(x(k)|z^k)$ is a Gaussian distribution, whose mean $\hat{x}(k|k)$ and covariance $P(k|k)$ are simply determined by (5.10)-(5.15).

Another distribution of interest, as we shall see later, is the probability density for the innovation process $\{z(k) - H(k)\hat{x}(k|k-1), k = 0,1,\ldots,K\}$. This process is a zero-mean White Gaussian Noise with a covariance B^k equal to

$$B^k = H(k)P(k|k-1)H^T(k) + R(k) \tag{5.16}$$

The Kalman Filtering Equations provide directly the set of estimates $\{\hat{x}(k|k),\ k = 0,1,\ldots,K\}$ through equations (5.10)-(5.15) by implicitly computing the probability density $P(x(k)|z^k)$. This is the optimal recursive implementable solution to our filtering problem.

We next discuss the smoothing problem which uses additional measurement data to compute the set of estimates.

5.4 The Smoothing Problem

The objective in the smoothing problem is to find an MAP estimate of the state sequence $x^K = (x(0),\ldots,x(K))$ based on all available measurement data $z^K = (z(0),\ldots,z(K))$.

The solution to this problem is given by the sequence $\hat{x}^{K|K} = \{\hat{x}(k|K),\ k = 0,1,\ldots,K\}$ for which $P(x^K|z^K)$ is maximum. In short, the problem is:

$$\text{find } \hat{x}^{K|K} \text{ such that } \forall\ x^K \in X^K,\ P(\hat{x}^{K|K}|z^K) \geq P(x^K|z^K)\ .$$

A straightforward computation of $\hat{x}^{K|K}$ would consist first in the determination of $P(x^K|z^K)$, then in the optimization of $P(x^K|z^K)$ over the super-space X^K.

In fact, because of the memoryless and Markov properties (5.1)-(5.2), we shall see that this estimation problem is equivalent to a "continuous" Viterbi algorithm, which requires much less computation.

For this purpose, let us define the following function at time k:

$$\tilde{P}(x(k)|z^k) = \sup_{x^{k-1} \in X^{k-1}} P(x^k|z^k) \qquad (5.17)$$

We denote by $\hat{x}[x(k)]$ the corresponding sequence terminating at $x(k)$. It is the equivalent of the survivor $\hat{d}[d(k)]$ in the Viterbi algo-

rithm, in the sense that the most likely sequence $\hat{x}^{K|K}$ must begin with the sequence $\hat{x}[\hat{x}(k|K)]$ for all $k = 0,1,\ldots,K$. This is true because of the following:

$$\begin{aligned}
&\forall\, x^k,\ P(x^k|z^k) \le \bar{P}(x(k)|z^k) = P(\hat{x}(x(k))|z^k) \\
&\Rightarrow \forall\, x^k \in X^K,\ \forall\, z(k+1),\ldots,z(K), \\
&\qquad P(x^K|z^K) \le P(\hat{x}(x(k)),\ x(k+1),\ldots,x(K)|z^k) \text{ using (5.1)-(5.2)} \\
&\Rightarrow \sup_{x^K \in X^K} P(x^K|z^K) \le \sup_{x(k),\ldots,x(K)\in X} P[\hat{x}[x(k)],x(k+1),\ldots,x(K)|z^K] \\
&\Rightarrow \sup_{x^K \in X^K} P(x^K|z^K) = \sup_{x(k),\ldots,x(K)\in X} P[\hat{x}(x(k)),x(k+1),\ldots,x(k)|z^K]
\end{aligned} \tag{5.18}$$

But $P(x^K|z^K)$ is maximum for $\{\hat{x}(k|K),\ k = 0,1,\ldots,k\}$. Assuming a single maximum for the density $P(x^K|z^K)$, we have: $\hat{x}[\hat{x}(k|K)] = (\hat{x}(0|k),\ldots,\hat{x}(k|K))$ using (4.18). Q.E.D.

Thus, for all $k = 0,1,\ldots,K$,

$$\hat{x}[\hat{x}(k|K)] = (\hat{x}(0|K),\ldots,\hat{x}(k|K)) \quad , \tag{5.19}$$

that is, we need only to store the sequence $\hat{x}[x(k)]$ and $\bar{P}(x(k)|z^k)$ at time k.

Using (5.1)-(5.2), we can determine $\bar{P}(x(k)|z^k)$ recursively from the following equation:

$$\bar{P}(x(k)|z^k) = \frac{P(z(k)|x(k))}{P(z(k)|z^{k-1})} \times \sup_{x(k-1)\in X} P(x(k)|x(k-1)) \times \bar{P}(x(k-1)|z^{k-1}) \tag{5.20}$$

where $P(z(k)|z^{k-1})$ is defined by (5.7)-(5.8) and $\bar{P}(x(0)|z^0) = P(x(0)|z(0))$ is the initial condition.

From this, we can compute the functions $\hat{x}[x,k]$ recursively. The recursion proceeds as follows: $\hat{x}[x,k]$ is extended by one time unit and the corresponding $\bar{P}(x(k),x(k+1)|z^{k+1}) \triangleq \frac{P(z(k+1)|x(k+1))}{P(z(k+1)|z^k)} \times P(x(k+1)|x(k)) \times \bar{P}(x(k)|z^k)$ is computed for all $x(k)$, $x(k+1)$, using the received measurement $z(k+1)$ at time $k+1$. Then the maximization of $\bar{P}(x(k+1),x(k)|z^{k+1})$ over $x(k) \in X$ leads to $\bar{P}(x(k+1)|z^{k+1})$ according to (5.20). The corresponding $\hat{x}_k[x(k+1)]$, that maximizes $\bar{P}(x(k+1),x(k)|z^{k+1})$, is used for the computation of $\hat{x}[x(k+1)]$, since we have:

$$\hat{x}[x(k+1)] = (\hat{x}[\hat{x}_k(x(k+1))], x(k+1)) \tag{5.21}$$

At time K, the maximization of $\bar{P}(x(K)|z^K)$ over X yields $\hat{x}(K|K)$ and subsequently the MAP estimate $\hat{x}^{K|K}$ of the sequence x^K, since, according to (5.19), we have:

$$\hat{x}[\hat{x}(K|K)] = \{\hat{x}(k|K),\ k = 0,1,\ldots,K\} .$$

The algorithm can be stated formally as follows:

STEP 0:
$$\bar{P}(x(0)|z^0) = P(x(0)|z(0))$$
$$\hat{x}[x(0)] = x(0)$$

STEP k
$$\bar{P}(x(k)|z^k) \quad , x(k) \in X$$
$$\hat{x}[x(k)] \quad , x(k) \in X$$

Compute $\bar{P}(x(k),x(k+1)|z^{k+1}) \triangleq \frac{P(z(k+1)|x(k+1))}{P(z(k+1)|z^k)} \times$

$$P(x(k+1)|x(k)) \times \bar{P}(x(k)|z^k) \quad ,$$

for all $x(k)$, $x(k+1)$.

Find $\bar{P}(x(k+1)|z^{k+1}) = \sup_{x(k)\in X} \bar{P}(x(k),x(k+1)|z^{k+1})$

Store $\bar{P}(x(k+1)|z^{k+1})$ and the corresponding $\hat{x}_k(x(k+1))$.

Then $\hat{x}[x(k+1)] = (\hat{x}[\hat{x}_k(x(k+1))], x(k+1))$

$k = k+1$ and repeat until $k = K$.

STEP K: $\bar{P}(x(K)|z^K)$, $x(K) \in X$

$\hat{x}[x(K)]$, $x(K) \in X$

Find $\sup_{x(K)\in X} \bar{P}(x(K)|z^K)$. The corresponding $x(K)$ is equal to $\hat{x}(K|K)$ and $\{\hat{x}(k|K), k = 0,1,\ldots,K\} = \hat{x}[\hat{x}(K|K)]$.

END.

This algorithm can be considered as the conceptual extension of the Viterbi algorithm to the continuous case: at each time k, we need only to store the "survivor" $\hat{x}[x(k)]$ and the function $\bar{P}(x(k)|z^k)$.

Of course, the main difficulty of this method lies in the explicit determination of $\bar{P}(x(k)|z^K)$ and $\hat{x}[x(k)]$. In the general case, such a determination is usually impossible. However, as in the filtering problem, the Linear Gaussian assumption introduces a great simplification in our smoothing problem with the Linear-Gaussian Smoothing Equations.

5.5 The Linear-Gaussian Smoothing Equations

Under the Linear-Gaussian assumption (5.3)-(5.5), the a posteriori probability distribution $P(x^K|z^K)$ is Gaussian. Thus, the MAP estimate of x^K, which is the state sequence $\hat{x}^{K|K} = \{\hat{x}(k|K), k = 0,1,\ldots,K\}$ for which $P(x^K|z^K)$ is maximum, is simply the conditional mean $E\{x^K|z^K\} = (E\{x(k)|z^K\}, k = 0,.,\ldots,K)$.

Meditch in [3] gives a method for computing recursively the set of smoothed estimates $\{\hat{x}(k|K), k = 0,1,\ldots,K\}$ and the corresponding covariances $\{P(k|K), k = 0,1,\ldots,K\}$.

The Linear-Gaussian Smoothing Equations are the following: for $k = K-1, K-2, \ldots, 0$,

$$\hat{x}(k|K) = \hat{x}(k|k) + L(k)[\hat{x}(k+1|K) - \hat{x}(k+1|k)] \tag{5.22}$$

$\hat{x}(K|K)$ is the boundary condition for $k = K-1$.

$$L(k) = P(k|k)F^T(k)P^{-1}(k+1|k) \tag{5.23}$$

$$P(k|K) = P(k|k) + L(k)[P(k+1|K) - P(k+1|k)]L^T(k) \tag{5.24}$$

$\hat{x}(k|k)$, $\hat{x}(k+1|k)$, $P(k|k)$, $P(k+1|k)$ are given by the Kalman Filtering Equations (5.10)-(5.15).

The filtering equations, as we saw in Section 3, compute recursively <u>forwards in time</u> the set of estimates $\{\hat{x}(k|k), k = 0,1,\ldots,K\}$ and the corresponding covariances $\{P(k|k), k = 0,1,\ldots,K\}$. Then the smoothing equations (5.22)-(5.24) compute recursively <u>backwards in time</u> the set of smoothed estimates $\{\hat{x}(k|K), k = K-1,K-2,\ldots,0\}$ and the corresponding covariances $\{P(k|K), k = K-1,\ldots,0\}$.

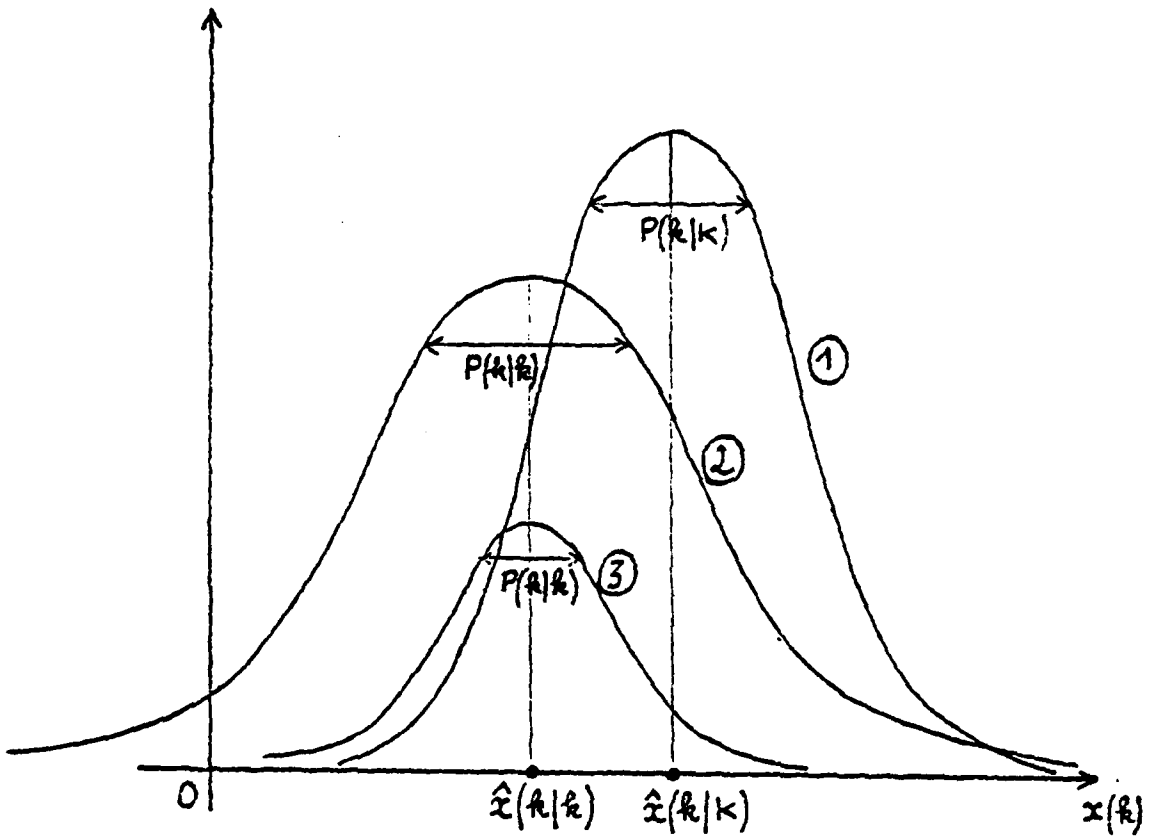
Thus, in the Linear Gaussian case, the smoothing equations simply refine the set of estimates given by the filtering equations.

The set of smoothed estimates is computed directly without determining explicitly the functions $\bar{P}(x(k)|z^k)$ for $k = 0,1,\ldots,K$. Later in this work, an expression for $\bar{P}(x(k)|z^k)$ will be needed. The probability distribution $P(x^k|z^k)$ is equal to $N[x^k - \hat{x}^{k|k}, P^{k|k}]$. It can be shown that $\bar{P}(x(k)|z^k) = \sup_{x^{k-1} \in X^{k-1}} P(x^k|z^k)$ is equal to:

$$\bar{P}(x(k)|z^k) = \sqrt{\frac{(2\pi)^n |P(k|k)|}{(2\pi)^{nk} |P^{k|k}|}} \times N[x(k) - \hat{x}(k|k), P(k|k)], \tag{5.25}$$

where $\hat{x}(k|k)$, $P(k|k)$ are given by the Kalman Filtering Equations (5.10)-(5.15) and $P^{k|k}$ is the covariance of the overall state sequence x^k. (See

FIGURE 5.1



$$\textcircled{1}\quad P(x(k)\,|\,z^K) = N\left[x(k)-\hat{x}(k|K),\ P(k|K)\right]$$

$$\textcircled{2}\quad P(x(k)\,|\,z^k) = N\left[x(k)-\hat{x}(k|k),\ P(k|k)\right]$$

$$\textcircled{3}\quad \bar{P}(x(k)\,|\,z^k) = \sqrt{\frac{(2\pi)^m\,|P(k|k)|}{(2\pi)^{m\hat{x}}\,|P^{\hat{x}}|}}\times P(x(k)\,|\,z^k)$$

Fig. 5.1).

5.6 *Summary*

In this chapter, we have discussed the MAP continuous state estimation problem. The difficulty in the filtering problem lies in the determination of the density $P(x(k)|z^k)$. However, under a Linear-Gaussian assumption, the set of estimates $\{\hat{x}(k|k),\ k = 0,1,\ldots,K\}$ can be computed directly.

The difficulty in the smoothing problem, which can be viewed as a "continuous" Viterbi algorithm, lies also in the determination of some functions $\bar{P}(x(k)|z^k)$ and $\hat{x}[x(k)]$. Again, the Linear-Gaussian assumption introduces a great simplification, in the sense that a direct recursive computation of the set of smoothed estimates $\{\hat{x}(k|K),\ k = K-1,\ldots,0\}$ becomes possible. Besides, this set of estimates is simply obtained by "refining" the set of filtered estimates.

Now, having presented the Viterbi algorithm in Chapter 4 and the MAP continuous state estimation problem in this chapter, we discuss in the next chapter the combined MAP hybrid state estimation problem.

CHAPTER 6

STATE ESTIMATION OF A HYBRID MARKOV PROCESS

In this chapter, we consider a single hybrid system. The purpose of this part is to obtain an algorithm for estimating the hybrid state sequence, that keeps the amount of computation at a reasonable level. The combined MAP hybrid state estimation algorithm in a blend of the Viterbi algorithm of Chapter 4 and the MAP continuous state estimation algorithm of Chapter 5. Depending on two possible choices of objective for this problem, there are two different formulations, as we shall next see.

6.1 Problem Formulations

The model for the hybrid system considered is the one of Chapter 2 Section 2. The system is observed during an interval [0,K]. The problem is to estimate, from the available measurement data $z^K = (z(0),\ldots, z(K))$, the hybrid state sequence $s^K = (s(0),\ldots,s(K))$.

The MAP estimation of s^K based on z^K is one possible objective. In this case, the problem is to find $\hat{s}^K = (\hat{d}^K,\hat{x}^K)$ such that:

$$\forall\, x^K \in X^K, \quad \forall\, d^K \in D^K, \quad P(\hat{x}^K,\hat{d}^K \mid z^K) \geq P(x^K,d^K \mid z^K) .$$

However, in some cases, more importance is given to the estimation of the discrete state sequence d^K. For instance, in failure detection problems, when the system has sufficient hardware back-up capabilities, we could be interested only in detecting the failures, without directly desiring to estimate the continuous state of the system. Similarly, in some surveillance problems, we can be more interested in the number of

targets in a certain area, than in their precise locations.

In these cases, we first perform a MAP estimation of the discrete state sequence d^K: so, the problem is to find d^K such that:

$$\forall\, d^K \in D^K\ ,\quad P(\hat{d}^K | z^K) \geq P(d^K | z^K)$$

Then, if we are still interested in the continuous state of the system, an MAP estimation of the sequence x^K can be done, conditioned on $\hat{d}^K$; this problem is simply the continuous state estimation problem discussed in Chapter 5. In this objective, we minimize the probability of an error in estimating the discrete state sequence d^K.

Both objectives do not yield necessarily the same estimates $(\hat{d}^K, \hat{x}^{K|K})$. The first approach leads to a better estimate of the pair (d^K, x^K), and the second one to a better estimate of the discrete sequence d^K.

In both cases, a graph can be associated with our Markov process, as in the Viterbi algorithm. In this graph, each node corresponds to a distinct discrete state d(k) of the system, and each branch represents a transition to some new discrete state at the next instant of time. The graph begins at a known discrete state d_0. Thus, to every possible discrete state sequence d^K, there corresponds a unique path through the graph, and vice versa.

In the next two sections, we consider both approaches, and derive for each case some pruning rules in the graph that are available to reduce the amount of computation.

6.2 Discrete State Estimation Algorithm

As mentioned above, under this objective we have to perform a MAP estimation of the discrete state sequence d^K: find d^K such that:

$$\forall\, d^K \in D^K\ ,\quad P(\hat{d}^K | z^K) \geq P(d^K | z^K)$$

A straightforward computation of d^K is combinatorial: find the most likely state sequence d^K among N^K state sequences. A method has to be found for decreasing the amount of computation.

This problem is formally equivalent to the problem of finding the shortest path in the above graph, if we define the length of $\lambda(d^k)$ of a path corresponding to the state sequence d^k by:

$$\lambda(d^k) = -\ell n\ P(d^k, z^k) \tag{6.1}$$

However, the solution to this shortest path problem is not as simple as for the discrete state case. The Viterbi algorithm uses the Markov and memoryless properties to keep only the N survivors $\hat{d}[d(k)]$ at each time $k = 0,1,\ldots,K$. But here, the memoryless property does not hold anymore: the probability distribution $P(z(k) | z^{k-1}, d^k)$ depends on the whole discrete state sequence d^k and on all the past measurements z^{k-1} through the continuous states of the hybrid system. Thus, the condition $\lambda(d_1^k) \leq \lambda(d_2^k)$, where $d_1(k) = d_2(k)$, is not sufficient anymore for discarding d_2^k at time k. As later measurements are received, d_2^k can well be a part of the most likely sequence d^K.

A stronger condition is needed for discarding a path corresponding to d^k at time k, without increasing the probability of an error in estimating the whole sequence d^K. This condition is a consequence of the following theorem:

Theorem 1: Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$. If for all $x(k) \in X$, $P(x(k), d_1^k | z^k) \leq P(x(k), d_2^k | z^k)$, then for all $z(k+1),\ldots,z(K)$ and for all $d(k+1),\ldots,d(K)$;

$P(d_1^K|z^K) \leq P(d_2^K|z^K)$ where $d_1^K = (d_1^k, d(k+1), \ldots, d(K))$ and $d_2^K = (d_2^k, d(k+1), \ldots, d(K))$.

Proof: The proof of this theorem is straightforward using the following recursive relation on $P(x(k), d^k|z^k)$:

$$P(x(k),d^k|z^k) = \frac{P(z(k)|d(k),x(k))}{P(z(k)|z^{k-1})} \times P(x(k),d^k|z^{k-1}) \qquad (6.2)$$

$$P(z(k)|z^{k-1} = \sum_{\{d^k\}} \int_X P(x(k),d^k|z^{k-1}) \times$$

$$P(z(k)|x(k), d(k))dx(k) \qquad (6.3)$$

$$P(x(k),d^k|z^{k-1}) = \int_X P(x(k),d(k)|x(k-1),d(k-1) \times$$

$$P(x(k-1),d^{k-1}|z^{k-1})dx(k-1) \qquad (6.4)$$

$$P(x(0),d^0|z^{-1}) \triangleq P(x(0)) \text{ is the initial condition} \qquad (6.5)$$

Thus, simply using (6.2), if $\forall\, x(k)$, $P(x(k),d_1^k|z^k) \leq P(x(k),d_2^k|z^k)$ where $d_1(k) = d_2(k)$, then $\forall\, x(k+1)$, $\forall\, d(k+1)$, $\forall\, z(k+1)$, $P(x(k+1),d_1^{k+1}|z^{k+1}) \leq P(x(k+1),d_2^{k+1}|z^{k+1})$. And proceeding by induction: $\forall\, x(K)$, $\forall\, d(k+1),\ldots,d(K)$, $\forall\, z(k+1),\ldots,z(K)$, $P(x(K), d_1^K|z^K) \leq P(x(k),d_2^K|z^K)$. Finally, by integrating on $x(K)$: $P(d_1^K|z^K) \leq P(d_2^K|z^K)$.

Q.E.D.

This theorem means that we can compare two discrete state sequences at time k, terminating at the same node d(k): if at this time, all the continuous states are less likely on one sequence and, if after that time we follow a same common path, then this trajectory will remain less likely, whatever the future measurements are. Therefore, we can

eliminate it immediately from consideration. Thus we have the first pruning rule in our graph:

Pruning Rule R1: Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$. If, for all x(k), $P(x(k), d_1^k | z^k) \leq P(x(k), d_2^k | z^k)$. (6.6), then the whole sequence d_1^k can be immediately discarded at time k, without increasing the probability of an error in estimating d^K.

The implementation of this pruning rule is not as simple as in the Viterbi algorithm. Besides, it is rarely possible to order each discrete state sequence using (6.6): most of the time, more than one "survivor" terminating at a given d(k) remain after having used this pruning rule. Therefore, we would like to extend the possibilities of eliminating some sequences. This is the subject of the following theorem, which allows comparisons between one sequence and a set of sequences:

Theorem 1': Let $d_0^k, d_1^k, \ldots, d_r^k$ be (r+1) discrete state sequences such that $d_0(k) = d_1(k) = \ldots = d_r(k)$. Let $(\alpha_i)_{1 \leq i \leq r}$ be r real positive coefficients such that $\sum_{i=1}^{r} \alpha_i = 1$. If for all $x(k) \in X$, $P(x(k), d_0^k | z^k) \leq \sum_{i=1}^{r} \alpha_i P(x(k), d_i(k) | z^k)$, then for all z(k+1),...,z(K), for all d(k+1),..., d(K), $P(d_0^K | z^K) \leq \sup_{1 \leq i \leq r} P(d_i^K | z^K)$, where for all i = 0,1,...,r, $d_i^K = (d_i^k, d(k+1), \ldots, d(k))$

Proof: The proof of theorem 1' is very similar to the one of theorem 1 and is omitted.

We notice that the set of α_i's is arbitrary as long as $\alpha_i \geq 0$ and $\sum_{i=1}^{r} d_u = 1$. A procedure for choosing (α_i) could be:

$$\max_{\alpha_i} \inf_{x(k)\in X} \sum_{i=1}^{r} \alpha_i P(x(k), d_i^k | z^k) - P(x(k), d_0^k | z^k)$$

$$\text{s.t. } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{r} \alpha_i = 1 \quad .$$

If this Max/Min is positive, then we have: $P(d_0^K | z^K) \leq \sup_{1\leq i\leq r} P(d_i^K | z^K)$.

Anyway, from Theorem 1' we deduce immediately the following pruning rule:

Generalized Pruning Rule: GR1: Let $d_0^k, d_1^k, \ldots, d_r^k$ be (r+1) discrete state sequences such that $d_0(k) = \ldots = d_r(k)$. If it exists r positive coefficients α_i such that $\sum_{i=1}^{r} \alpha_i = 1$ and

$$\forall\, x(k) \in X \quad , \; P(x(k), d_0^k | z^k) \leq \sum_{i=1}^{r} \alpha_i P(x(k), d_i^k | z^k) \quad , \qquad (6.7)$$

then d_0^k can be immediately discarded at time k, without increasing the probability of an error in estimating d^K.

GR1 is a more performant pruning rule than R1, in the sense that more state sequences can be discarded at a given time using that rule (it reduces to R1 when r = 1); but it is also more difficult to implement.

Using R1 or GR1, we are now capable of discarding some sequences during the formation of the graph. Of course, since these rules are based on functional inequalities, more than one "survivor" will often remain for every sequence terminating at a given node. At time K, we choose $\hat{d}^K$ among the remaining state sequences. This is a substantial progress on the straightforward computation.

The algorithm can be stated formally as follows: let $\mathcal{D}^k$ be the set of remaining sequences at time k.

ALGORITHM 1

STEP 0: $\begin{cases} \mathcal{D}^0 = d_0 \\ P(x(0), d^0 | z^0) = P(x(0) | z(0), d_0) \end{cases}$

STEP k: $\begin{cases} \mathcal{D}^{k-1} \\ P(x(k-1), d^{k-1} | z^{k-1}) \text{ for all } d^{k-1} \in \mathcal{D}^{k-1},\ x(k-1) \in X \end{cases}$

Compute $P(x(k), d^k | z^k)$ using (6.2)-(6.5), for all $x(k) \in X$, $d^k \in \mathcal{D}^{k-1} \times D$.

Apply GR1 to d^k: for all $d_i^k \in \mathcal{D}^{k-1} \times D - \{d^k\}$,

$$\max_{\alpha_i} \inf_{x(k) \in X} \sum_i \alpha_i P(x(k), d_i^k | z^k) - P(x(k), d^k | z^k) \begin{matrix} \text{YES} \\ > \\ < \\ \text{NO} \end{matrix} 0$$

s.t. $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$

YES: Delete d^k

NO: Store d^k in $\mathcal{D}^k$

$k = k+1$ and repeat until $k = K$

STEP K: $\begin{cases} \mathcal{D}^K \\ P(x(K), d^K | z^K) \text{ for all } x(K) \in X,\ d^K \in \mathcal{D}^K \end{cases}$

Compute $P(d^K | z^K) = \int_X P(x(K), d^K | z^K) dx(K)$, for all $d^K \in \mathcal{D}^K$.

Find $\sup_{d^K \in \mathcal{D}^K} P(d^K | z^K)$. The corresponding d^K is equal to $\hat{d}^K$.

END.

If we are also interested in an estimate of the continuous se-

quence x^K, we have to combine Algorithm 1 with the MAP continuous state estimation algorithm of Chapter 5, and keep also track at time k of $\bar{P}(x(k),d^k|z^k)$ and $\hat{x}[x(k),d^k]$.

Of course, the difficulty in algorithm 1 lies in the explicit determination of $P(x(k),d^k|z^k)$ and in the application of GR1. The Linear-Gaussian assumption introduces a great simplification in these computations, as we shall see in Chapter 7. We next discuss the second objective of a hybrid state estimation problem.

6.3 Hybrid State Estimation Algorithm

The objective in this part is to find the MAP estimate of the hybrid state sequence s^K based on all available measurement data. The solution to this problem is given by $\hat{s}^K = (\hat{d}^K,\hat{x}^{K|K})$ such that:

$$\forall\, d^K \in D^K,\quad \forall\, x^K \in X^K\ ,\ P(\hat{d}^K,\hat{x}^{K|K}|z^K) \geq P(d^K,x^K|z^K)\quad .$$

This straightforward computation of s^K is performed as follows: first, on each sequence d^K, the problem is to find the *continuous* state sequence $\hat{x}^{K|K}(d^K) = (\hat{x}_{0|K}(d^K),\ldots,\hat{x}_{K|K}(d^K))$ for which $P(x^K|d^K,z^K)$ is maximum: this is the smoothing problem, conditioned on d^K, which has been presented in Chapter 5. Then, the estimation of d^K consists in finding the sequence $\hat{d}^K$ such that:

$$\forall\, d^K \in D^K,\ P(\hat{d}^K,\hat{x}^{K|K}(\hat{d}^K)|z^K) \geq P(d^K,\hat{x}^{K|K}(d^K)|z^K) \qquad (6.8)$$

The evaluation of $\hat{d}^K$ is strictly combinatorial: find the sequence $\hat{d}^K$ verifying (6.8) among N^K state sequences. So, as for the Viterbi algorithm, we have to find a way for reducing the amount of computation.

This problem is equivalent to the one of finding the shortest path in the graph considered in Section 1, if we define the length $\lambda(d^k)$ of a path corresponding to the state sequence d^k by:

$$\lambda(d^k) = - \ell n \sup_{x^k \in X^k} P(x^k, d^k | z^k) \tag{6.9}$$

As for the first objective, the relation $\lambda(d_1^k) \leq \lambda(d_2^k)$ where $d_1(k) = d_2(k)$, is not sufficient for discarding d_2^k at time k, since it does <u>not</u> imply that for all $z(k+1),\ldots,z(K)$, and for all $d(k+1),\ldots,d(K)$, $\lambda(d_1^K | z^K) \leq \lambda(d_2^K | z^K)$.

A stronger condition is needed for discarding optimally a path corresponding to a state sequence d^k at time k. This condition is based on the recursively computable functions $\tilde{P}(x(k), d^k | z^k) = \sup_{x^{k-1} \in X^{k-1}} P(x^k, d^k | z^k)$

It is a consequence of the following theorem:

<u>Theorem 2</u>: Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$. If, $\forall\, x(k) \in X$, $\tilde{P}(x(k), d_1^k | z^k) \leq \tilde{P}(x(k), d_2^k | z^k)$, then $\forall\, z(k+1),\ldots,z(K)$, $\forall\, d(k+1),\ldots,d(K)$, $\sup_{x^k \in X^k} P(x^K, d_1^K | z^K) \leq \sup_{x^K \in X^K} P(x^K, d_2^K | z^K)$ where $d_1^K = (d_1^k, d(k+1),\ldots,d(K))$ and $d_2^K = (d_2^k, d(k+1),\ldots,d(K))$.

<u>Proof</u>: The proof of this theorem is straightforward using the following recursive relations on $\tilde{P}(x(k), d^k | z^k)$,

$$\tilde{P}(x(k), d^k | z^k) = \frac{P(z(k) | d(k), x(k))}{P(z(k) | z^{k-1})} \times \sup_{x(k-1) \in X} P(x(k), d(k) | x(k-1), d(k-1)) \times P(x(k-1) d^{k-1} | z^{k-1}) \tag{6.10}$$

$P(z(k)|z^{k-1})$ is defined by (6.3)-(6.4).

$\bar{P}(x(0),d^0|z^0) \triangleq P(x(0)|z(0), d_0)$ is the initial condition (6.11)

Now, simply using (6.10), we have:

$\forall\, x(k) \in X,\ \bar{P}(x(k),d_1^k|z^k) \leq \bar{P}(x(k),d_2^k|z^k)$ where $d_1(k) = d_2(k)$

$\Rightarrow \forall\, x(k+1),\ \forall\, z(k+1),\ \forall\, d(k+1),\ \bar{P}(x(k+1),d_1^{k+1}|z^{k+1}) \leq$

$\bar{P}(x(k+1),d_2^{k+1}|z^{k+1})$

Proceeding by induction, we have:

$\forall x(K);\ \forall z(k+1),\ldots,z(K);\ \forall\, d(k+1),\ldots,d(K),$

$\bar{P}(x(K),d_1^K|z^K) \leq \bar{P}(x(K),d_2^K|z^K)$

$\Rightarrow \forall\, z(k+1),\ldots,z(K);\ \forall\, d(k+1),\ldots,d(K)\ ,$

$\sup_{x^K \in X^K} P(x^K,d_1^K|z^K) \leq \sup_{x^K \in X^K} P(x^K,d_2^K|z^K)$

Q.E.D.

As for the first objective, we can compare two sequences terminating at the same state d(k). The comparison is based on $\bar{P}(x(k),d^k|z^k)$ instead of $P(x(k),d^k|z^k)$, since to delete an hypothesis d_1^K at time K, we need to have $\sup_{x^K \in X^K} P(x^K,d_1^K|z^K) \leq \sup_{x^K \in X^K} P(x^K,d_2^K|z^K)$ and <u>not</u> $P(d_1^K|z^K) \leq P(d_2^K|z^K)$.

From Theorem 2, we deduce immediately the first pruning rule in our graph.

Pruning Rule R2: Let d_1^k and d_2^k be two state sequences such that $d_1(k) = d_2(k)$. If, for all $x(k) \in X$, $\bar{P}(x(k), d_1^k | z^k) \leq \bar{P}(x(k), d_2^k | z^k)$, (6.12) then d_1^k can be immediately discarded at time k.

Since it is not possible to order, using (6.12), all the sequences terminating at a given node d(k), most of the time, more than one "survivor" terminating at d(k) will remain. The possibilities for deleting a sequence are increased, if we compare this discrete sequence with a set of discrete sequences terminating at the same node. This is the subject of the following theorem:

Theorem 2': Let $d_0^k, d_1^k, \ldots, d_r^k$ be (r+1) discrete state sequences such that $d_0(k) = \ldots = d_r(k)$. If, for all $x(k) \in X$, $\bar{P}(x(k), d_0^k | z^k) \leq \sup_{1 \leq i \leq r} \bar{P}(x(k), d_i^k | z^k)$, then $\forall d(k+1), \ldots, d(K)$, $\forall z(k+1), \ldots, z(K)$, $\sup_{x^K \in X^K} P(x^K, d_0^K | z^K) \leq \sup_{1 \leq i \leq r} \sup_{x^K \in X^K} P(x^K, d_i^K | z^K)$ where, for all $i = 0,1,\ldots,r$, $d_i^K = (d_i^k, d(k+1), \ldots, d(K))$.

Proof: The proof is similar to the one of Theorem 2 and is omitted.

The generalized pruning rule 2 is a direct consequence of this theorem:

Generalized Pruning Rule GR2: Let $d_0^k, \ldots, d_r^k$ be (r+1) sequences such that $d_0(k) = \ldots = d_r(k)$. If for all $x(k) \in X$, $\bar{P}(x(k), d_0^k | z^k) \leq \sup_{1 \leq i \leq r} \bar{P}(x(k), d_i^k | z^k)$, then d_0^k can be immediately discarded at time k.

Using R2 or GR2, we can discard some sequences during the formation of the graph. At time K, we choose d^K among the sequences by:

$$P(\hat{d}^K, x^{MK}(\hat{d}^K) | z^K \geq P(d^K, x^{K|K}(d^K) | z^K) , \quad (6.13)$$

for all the remaining sequence d^K. The estimate of the hybrid state sequence is given by $\hat{s}^K = (\hat{d}^K, \hat{x}^{K|K}(\hat{d}^K))$. $\hat{x}^{K|K}(\hat{d}^K)$, which is the MAP estimate of x^K conditioned on $\hat{d}^K$, is determined by the continuous state estimation algorithm of Chapter 5. So, by keeping track, at each time k, not only of the functions $\bar{P}(x(k),d^k|z^k)$, but also of the survivors $\hat{x}[x(k),d^k]$ for all the remaining sequence d^k, we will be able to perform at time K the complete hybrid estimation.

The algorithm can be stated formally as follows: let $\mathcal{D}^k$ be the remaining sequences at time k.

ALGORITHM 2

STEP 0:
$$\begin{cases} \mathcal{D}^0 = d_0 \\ \bar{P}(x(0),d^0|z^0) = P(x(0)|z(0),d_0) \\ \hat{x}[x(0),d^0] = x(0) \end{cases}$$

STEP k:
$$\begin{cases} \mathcal{D}^{k-1} \\ \bar{P}(x(k-1),d^{k-1}|z^{k-1}) \quad \text{for all } x(k)\in X,\ d^{k-1} \in \mathcal{D}^{k-1} \\ \hat{x}[x(k-1),d^{k-1}] \end{cases}$$

Compute $\bar{P}(x(k),d^k|z^k)$ and $\hat{x}[x(k),d^k]$ for all $x(k) \in X$, $d^k \in \mathcal{D}^{k-1} \times D$, using (6.10).

Apply GR2 to d^k: for all $d_i^k \in \mathcal{D}^{k-1} \times D - \{d^k\}$,
$\forall\, x(k) \in X$, $\bar{P}(x(k),d^k|z^k) \underset{\text{NO}}{\overset{\text{YES}}{\lessgtr}} \sup_i \bar{P}(x(k),d^k|z^k)$

YES: Delete d^k

NO: Store d^k in $\mathcal{D}^k$

$k = k+1$ and repeat until $k = K$

STEP K:
$$\begin{cases} \mathcal{D}^K \\ \tilde{P}(x(K), d^K | z^K) \quad \text{for all } x(K) \in X,\ d^K \in \mathcal{D}^K \\ \hat{x}[x(K), d^K] \end{cases}$$

Compute $P(\hat{x}^{K|K}(d^K), d^K | z^K) = \sup_{x(K) \in X} \tilde{P}(x(K), d^K | z^K)$

Find $\sup_{d^K \in \mathcal{D}^K} P(\hat{x}^{K|K}(d^K), d^K | z^K)$. The corresponding $(\hat{d}^K, \hat{x}^{K|K}(\hat{d}^K))$ is equal to $\hat{\xi}^K$. END.

The difficulty in an actual implementation of Algorithm 2 lies in an explicit determination of $\tilde{P}(x(k), d^k | z^k)$ and $\hat{x}[x(k), d^k]$, in the storage of these functions, and in the application of GR2, which is based on a functional inequality. Once again, the Linear-Gaussian assumption introduces a great simplification as we shall see in Chapter 7.

6.4 Summary and Conclusion

In this chapter, we have discussed the state estimation of a hybrid Markov process. Depending on the application, we can be interested either in estimating the hybrid state sequence, or in estimating only the discrete state sequence of this system. In both cases, we have shown the formal equivalence between our estimation problem and a shortest path problem in a certain graph. Then we have derived some pruning rules, based on the distributions $P(x(k), d^k | z^k)$ or $\tilde{P}(x(k), d^k | z^k)$, that are used for the deletion of some discrete state sequences during the formation of the graph. This adaptive estimation algorithm should decrease substantially the amount of computation required at time K.

In addition to an explicit determination of the above distributions, there is also the problem of implementing in a simple way these pruning rules. This can be done under an additional Linear-Gaussian assumption, as we shall see in the next chapter.

CHAPTER 7

STATE ESTIMATION OF A HYBRID GAUSS-MARKOV PROCESS

The purpose of this chapter is to apply the results of Chapter 6 to the special case of a Gauss-Markov process. The system model is the one of Chapter 2 Section 2. The additional Linear-Gaussian assumption yields, as in Chapter 5, to an explicit determination of the distributions $P(x(k),d^k|z^k)$ and $\bar{P}(x(k),d^k|z^k)$ of the last chapter. Furthermore, under this assumption, pruning rules R1 and R2 can be implemented in a simple way. Finally, an additional deletion technique, specific to Gauss-Markov processes, can be found.

7.1 Computation of some densities under a Linear-Gaussian assumption

Under the Linear-Gaussian assumptions (2.4)-(2.6), $P(x(k),d^k|z^k)$ and $\bar{P}(x(k),d^k|z^k)$, are normal functions, weighted by some coefficients. Here, we derive how these functions can be easily computed recursively.

We have the relation:

$$P(x(k),d^k|z^k) = P(x(k)|d^k,z^k) \times P(d^k|z^k) \qquad (7.1)$$

The distribution $P(x(k)|d^k,z^k)$ represents the conditional a posteriori density for the continuous state $x(k)$ on a discrete state sequence $d^k = (d(0),\ldots,d(k))$. According to Section 3 of Chapter 5, we know that $P(x(k)|d^k,z^k)$ is a Gaussian density. Its mean $\hat{x}_{k|k}(d^k)$ and covariance $P_{k|k}(d^k)$ depend on the discrete sequence d^k through equations (2.4)-(2.6). They are given recursively by the following equations,

obtained directly from the Kalman Filtering Equations (5.10)-(5.15). Here, the state transition, observation, and covariance matrices at time k depend also on the discrete state values d(k) = p, d(k+1) = q at time k and k+1:

$$\hat{x}_{k|k}(d^k) = \hat{x}_{k|k-1}(d^k) + K_k(d^k)[z(k) - H(p,k)\hat{x}_{k|k-1}(d^k)] \quad (7.2)$$

$$\hat{x}_{k+1|k}(d^k) = F(p,q,k)\hat{x}_{k|k}(d^k) \quad (7.3)$$

$$P_{k|k}(d^k) = [I - K_k(d^k)H(p,k)]P_{k|k-1}(d^k) \quad (7.4)$$

$$P_{k+1|k}(d^{k+1}) = F(p,q,k)P_{k|k}(d^k)F^T(p,q,k) + Q(p,q,k) \quad (7.5)$$

$$K_k(d^k) = P_{k|k-1}(d^k)\,(p,k)[H(p,k)P_{k|k-1}(d^k)H^T(p,k) + R(p,k)]^{-1} \quad (7.6)$$

$\hat{x}_{0|-1}(d_0) = 0$ and $P_{0|-1}(d_0) = P_o$ is the initial condition (7.7)

$F(p,q,k)$, $G(p,k)$, $Q(p,q,k)$, $R(p,k)$, P_0 are defined by (2.4)-(2.6).

The coefficient in front of the Gaussian density is equal to: $P(d^k|z^k) = P(d^k,z^k)/P(z^k)$ where $P(z^k)$ is only a normalization term. $P(d^k,z^k)$ can be determined recursively from:

$$P(d^k,z^k) = P(z(k)|z^{k-1},d^k) \times P(d(k)|d(k-1)) \times P(d^{k-1},z^{k-1}). \quad (7.8)$$

$P(d(k)|d(k-1))$ is known, and, as we saw in Chapter 5, $P(z(k)|z^{k-1},d^k)$ is the probability density for the innovation process, conditioned on

d^k. Therefore we have:

$$P(z(k)|z^{k-1},d^k) = N[z(k) - H(p,k)\hat{x}_{k|k-1}(d^k),\ B(d^k)] \qquad (7.9)$$

where

$$B(d^k) = H(p,k)P_{k|k-1}(d^k)H^T(p,k) + R(p,k). \qquad (7.10)$$

Thus, $P(x(k),d^k|z^k)$ can be easily computed recursively using (7.1)-(7.10).

Similarly, we have:

$$\tilde{P}(x(k),d^k|z^k) = \tilde{P}(x(k)|d^k,z^k) \times P(d^k|z^k) \qquad (7.11)$$

According to Section 5 of Chapter 5, $\tilde{P}(x(k)|d^k,z^k)$ is a Gaussian density, conditioned on d^k, weighted by a coefficient, and whose mean and covariance are $\hat{x}_{k|k}(d^k)$ and $P_{k|k}(d^k)$ are given by (7.2)-(7.7):

$$\tilde{P}(x(k)|d^k,z^k) = \sqrt{\frac{(2\pi)^n|P_{k|k}(d^k)|}{(2\pi)^{nk}|P^{k|k}(d^k)|}} \times N(x(k) - \hat{x}_{k|k}(d^k),$$

$$P_{k|k}(d^k)], \qquad (7.12)$$

$P^{k|k}(d^k)$ is the covariance of the whole state sequence x^k, conditioned on d^k. The computation of $P^{k|k}(d^k)$ can be done as follows: the diagonal "terms" in $P^{k|k}(d^k)$ are equal to $P_{i|k}(d^k)$, the smoothed covariances defined in Chapter 5 by (5.23)-(5.24); the off-diagonal matrices $P_{i+j,i}$ corresponding to the $(i+j)^{th}$ row block and i^{th} comumn block are equal to:

$$P_{i+j,i} = F(d(j-1),d(j),j-1) \times \ldots \times F(d(i),d(i+1),i)P_{i|k}(d^k)$$

$$+(F(d(j-1),d(j),j-1) \times \ldots \times F(d(i),d(i+1),i)\hat{x}_{i|k} -$$

$$- \hat{x}_{i+j|k})\hat{x}_{i|k}^T \ . \qquad (7.13)$$

Thus, $\tilde{P}(x(k),d^k|z^k)$ can be obtained from (7.2)-(7.13).

These functions are used for the pruning rules of Algorithm 1 and 2 of the last chapter. The Linear-Gaussian assumption yields a simple implementation of some of these rules. This is the subject of the following section.

7.2 Actual Implementation of R1 and R2 under the Linear-Gaussian Assumption

Pruning rules R1 and R2 of the last chapter are based on functional inequalities between the above distributions, $P(x(k),d^k|z^k)$ for R1 and $\tilde{P}(x(k),d^k|z^k)$ for R2. It is straightforward to find a simple test equivalent to the functional inequality between two Gaussian distributions, weighted by some coefficients. This is the subject of the following Lemma:

Lemma: Let $f_1(x) = \alpha_1 \exp -\{\frac{(x-x_1)^T B_1^{-1}(x-x_1)}{2}\}$ and $f_2(x) = \alpha_2 \exp -\{\frac{(x-x_2)^T B_2^{-1}(x-x_2)}{2}\}$ be two functions defined over a state space X, such that $0 < B_1$ and $0 < B_2$. Then, for all $x \in X$,

$$f_1(x) \leq f_2(x) \iff \begin{cases} 0 < B_1 < B_2 \\ (x_1-x_2)^T(B_2-B_1)^{-1}(x_1-x_2) \leq 2 \ln \frac{\alpha_2}{\alpha_1} \end{cases}$$

Proof: The proof is left to Appendix B.

This Lemma yields a very simple implementation of pruning rule R1 and R2 in the Linear-Gaussian case:

Pruning Rule R1(L): Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$. Let $\hat{x}_{k|k}(d_1^k)$, $P_{k|k}(d_1^k)$, $\hat{x}_{k|k}(d_2^k)$, $P_{k|k}(d_2^k)$ be the corresponding means and covariances as defined by (7.2)-(7.7). Let

$$\beta(d_1^k,d_2^k) = (\hat{x}_{k|k}(d_2^k) - \hat{x}_{k|k}(d_1^k))^T(P_{k|k}(d_2^k) - P_{k|k}(d_1^k))^{-1}(\hat{x}_{k|k}(d_2^k) - \hat{x}_{k|k}(d_1^k)) + 2\,\ell n \frac{P(d_1^k|z^k)}{P(d_2^k|z^k)} + \ell n \frac{|P_{k|k}(d_2^k)|}{|P_{k|k}(d_1^k)|} \tag{7.14}$$

If $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$ and $\beta(d_1^k,d_2^k) \leq 0$, then d_1^k can be immediately discarded at time k, without increasing the probability of an error in estimating d^K.

Proof. The proof comes directly from pruning rule R1, the expression of $P(x(k),d^k|z^k)$ in the LG case and the applications of the Lemma.

We have also the following pruning rule R2(L), obtained directly from R2:

Pruning Rule R2(L): Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$. Let $\hat{x}_{k|k}(d_1^k)$, $P_{k|k}(d_1^k)$, $P^{k|k}(d_1^k)$, $\hat{x}_{k|k}(d_2^k)$, $P^{k|k}(d_2^k)$, be the corresponding means and covariances as defined by (7.2)-(7.13). Let also:

$$\gamma(d_1^k,d_2^k) = (\hat{x}_{k|k}(d_2^k) - \hat{x}_{k|k}(d_1^k))^T(P_{k|k}(d_2^k) - P_{k|k}(d_1^k))^{-1}(\hat{x}_{k|k}(d_2^k) - \hat{x}_{k|k}(d_1^k)) + 2\,\ell n \frac{P(d_1^k|z^k)}{P(d_2^k|z^k)} + \ell n \frac{|P^{k|k}(d_2^k)|}{|P^{k|k}(d_1^k)|} \tag{7.15}$$

If $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$ and $\gamma(d_1^k, d_2^k) \leq 0$, then d_1^k can be immediately discarded at time k, without increasing the probability of an error in estimating the whole hybrid state sequence $s^K = (d^K, x^K)$.

Proof: The proof is similar to the one of R1(L) and is omitted.

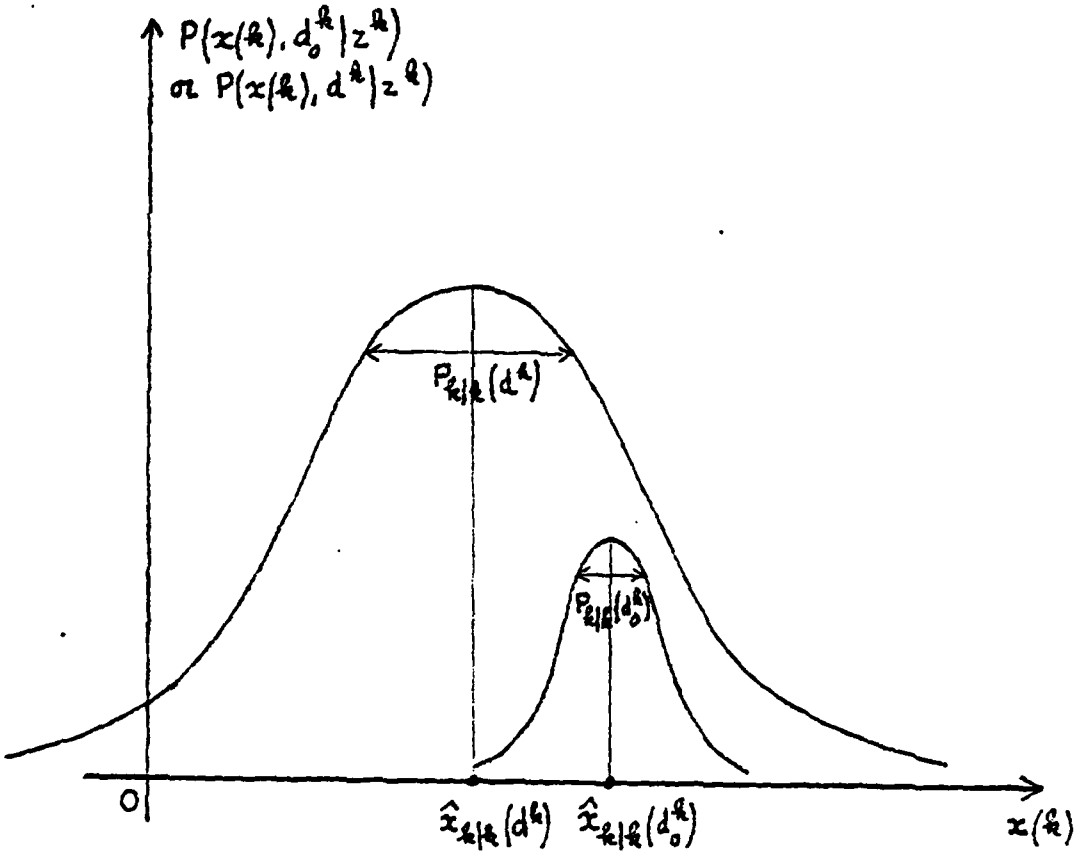
The condition $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$ means that we can only discard distributions with "small" covariances. Of course, a necessary condition for considering the deletion of d_1^k by d_2^k is that d_2^k be at least more likely at time k, i.e., $P(d_1^k|z^k) \leq P(d_2^k|z^k)$. Thus, if we want to discard a given sequence d_0^k, we try to compare it only with other more likely sequences d^k, whose associated Gaussian distributions have a larger covariance (see Fig. 7.1).

Unfortunately, no such simple tests can be found for the generalized pruning rules GR1 and GR2, which are certainly more powerful, in terms of the number of discrete state sequences, that could be deleted at a given time. For these rules, the conditions would have to be verified numerically.

So, there exists a tradeoff: complexity vs. performance in our algorithms. The need for an easy implementation of the pruning rules leads to the use of R1(L) or R2(L), which are less powerful than GR1 or GR2, in the sense that more state sequences will remain at time K.

Another interesting feature of the LG case is the following: the future values of the coefficients of comparison β or γ, between 2 state sequences following the same path from the current time, can be determined at this time, independently of the future measurements. This leads to some other sequence deletion opportunities, as we shall next see.

FIGURE 7.1



7.3 Another Pruning Rule Specific to the LG Case

Let us consider two state sequences d_1^k and d_2^k such that $d_1(k) = d_2(k)$ and $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$. If, according to our objective, $\beta(d_1^k, d_2^k) > 0$ or $\gamma(d_1^k, d_2^k) > 0$, then we cannot discard immediately d_1^k at time k, using pruning rule R1 or R2. However, the future values $\beta(d_1^{k'}, d_2^{k'})$ or $\gamma(d_1^{k'}, d_2^{k'})$ of the coefficients at time $k' > k$, where $d_1^{k'} = (d_1^k, d(k+1), \ldots, d(k'))$ and $d_2^{k'} = (d_2^k, d(k+1), \ldots, d(k'))$, are precomputable at time k. This is true because of the following theorem.

Theorem: Let d_1^k and d_2^k be two discrete state sequences such that $d_1(k) = d_2(k)$ and let $\beta(d_1^k, d_2^k)$ and $\gamma(d_1^k, d_2^k)$ be the corresponding coefficients as defined by (7.14)-(7.15). Assuming F(p,q,k) is invertible, then for all z(k+1), for all d(k+1),

$$\beta(d_1^{k+1}, d_2^{k+1}) = \beta(d_1^k, d_2^k) + \ell n \frac{|B(d_2^{k+1})|}{|B(d_1^{k+1})|} + \ell n \frac{|P_{k+1|k+1}(d_2^{k+1})|}{|P_{k+1|k+1}(d_1^{k+1})|} - \ell n \frac{|P_{k|k}(d_2^k)|}{|P_{k|k}(d_1^k)|} \tag{7.16}$$

$$\gamma(d_1^{k+1}, d_2^{k+1}) = \gamma(d_1^k, d_2^k) + \ell n \frac{|B(d_2^{k+1})|}{|B(d_1^{k+1})|} + \ell n \frac{|P^{k+1|k+1}(d_2^{k+1})|}{|P^{k+1|k+1}(d_1^{k+1})|} - \ell n \frac{|P^{k|k}(d_2^k)|}{|P^{k|k}(d_1^k)|} \tag{7.17}$$

Proof: The proof of this theorem is in Appendix C.

The main consequence of this theorem is that we can predict, through the coefficients β or γ, the future evolution in the relative position of two distributions, associated with a common sequence from the current

time, since the covariances are precomputable at time k.

Furthermore, using (7.15) and (7.16), we have the following inequalities:

$$\beta(d_1^{k+1},d_2^{k+1}) < \beta(d_1^k,d_2^k) \tag{7.18}$$

$$\gamma(d_1^{k+1},d_2^{k+1}) < \gamma(d_1^k,d_2^k) \tag{7.19}$$

So, starting at time k, the sequences $\beta(d_1^k,d_2^k)$, $\beta(d_1^{k+1},d_2^{k+1}),\ldots\beta(d_1^K,d_2^K)$ or $\gamma(d_1^k,d_2^k)$, $\gamma(d_1^{k+1},d_2^{k+1}),\ldots,\gamma(d_1^K,d_2^K)$, where for all $i \in [k,K]$, $d_1^i = (d_1^k,d(k+1),\ldots,d(i))$ and $d_2^i = (d_2^k,d(k+1),\ldots,d(i))$, are decreasing sequences. Even if we cannot discard d_1^k at time k using R1(L) or R2(L) because $\beta(d_1^k,d_2^k) > 0$ or $\gamma(d_1^k,d_2^k) > 0$, there is a chance that $\beta(d_1^{k'},d_2^{k'})$ or $\gamma(d_1^{k'},d_2^{k'})$ become negative at a later time k' that we are able to determine through (7.16)-(7.17). So, we can delete $d_1^{k'}$ immediately at time k, if the two following conditions are satisfied:

(i) $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$ (which implies $P_{k'|k'}(d_1^{k'}) < P_{k'|k'}(d_2^{k'})$)

(ii) $$\beta(d_1^k,d_2^k) \leq \sum_{j=k+1}^{k'} \ell n \frac{|B(d_1^j)|}{|B(d_2^j)|} + \ell n \frac{|P_{k'|k'}(d_1^{k'})|}{|P_{k'|k'}(d_2^{k'})|} + \ell n \frac{|P_{k|k}(d_2^k)|}{|P_{k|k}(d_1^k)|} \tag{7.20}$$

or

$$\gamma(d_1^k,d_2^k) \leq \sum_{j=k+1}^{k'} \ell n \frac{|B(d_1^j)|}{|B(d_2^j)|} + \ell n \frac{|P^{k'|k'}(d_1^{k'})|}{|P^{k'|k'}(d_2^{k'})|} + \ell n \frac{|P^{k|k}(d_2^k)|}{|P^{k|k}(d_1^k)|} \tag{7.21}$$

Furthermore, we can simply eliminate the state sequence d_1^k at time k, if for all the future sequences $d_1(k+1),\ldots,d_1(K)$, we can find at least one sequence $d_2^K = (d_2^k, d_1(k+1),\ldots,d_1(K))$ such that $P_{k|k}(d_1^k) < P_{k|k}(d_2^k)$ and (7.20) or (7.21) is satisfied for k' = K.

This constitutes the additional pruning rule in our graph. Let us state it formally for both objectives as follows:

Additional Pruning Rule AR1(L): Let d_0^k be a given state sequence up to time k. If, for all $d_0(k+1),\ldots,d_0(K)$, there exists a sequence $d^K = (d^k, d_0(k+1),\ldots,d_0(K))$ where $d(k) = d_0(k)$ and $P_{k|k}(d_0^k) < P_{k|k}(d^k)$ and

$$\gamma(d_0^k, d^k) \leq \sum_{j=k+1}^{K} \ell n \frac{|B(d_0^j)|}{|B(d^j)|} + \ell n \frac{|P_{K|K}(d_0^K)|}{|P_{K|K}(d^K)|} + \ell n \frac{|P_{k|k}(d^k)|}{|P_{k|k}(d_0^k)|}, \tag{7.22}$$

Then d_0^k can be immediately deleted at time k, without increasing the probability of an error in estimating d^K.

We have the following pruning rule for our second objective:

Additional Pruning Rule AR2(L): Let d_0^k be a given state sequence up to time k. If, for all $d_0(k+1),\ldots,d_0(K)$, there exists a sequence $d^K = (d^k, d_0(k+1),\ldots,d_0(K))$ where $d(k) = d_0(k)$ and $P_{k|k}(d_0^k) < P_{k|k}(d^k)$ and

$$\gamma(d_0^k, d^k) \leq \sum_{j=k+1}^{K} \ell n \frac{|B(d_0^j)|}{|B(d^j)|} + \ell n \frac{|P^{K|K}(d_0^K)|}{|P^{K|K}(d^K)|} + \ell n \frac{|P^{k|k}(d^k)|}{|P^{k|k}(d_0^k)|}, \tag{7.23}$$

then d_0^k can be immediately discarded at time k, without increasing the probability of an error in estimating the hybrid sequence s^K.

When the covariances tend quite rapidly (in a few steps) to their steady-state value, the coefficients β and γ tend also rapidly to a

steady-state value on a given path. This means, that in this case these coefficients must be positive but close to 0, in order for AR1(L) or AR2(L) to work effectively.

We illustrate AR1(L) in figure (7.2), for the case: d(k) can take only 2 values {0,1} at each time k and k = K-1.

Together with R1(L) and R2(L), these pruning rules lead to the following algorithms that are actually implementable versions of Algorithm 1 and 2 of the last chapter.

7.4 State Estimation Algorithms under the Linear-Gaussian Assumption

To apply R1(L) or R2(L), we need to compute, at each time k, the probability of each remaining discrete sequence at that time, its associated continuous state estimate and covariance; for AR1(L) and AR2(L), we need also to compute the future covariances.

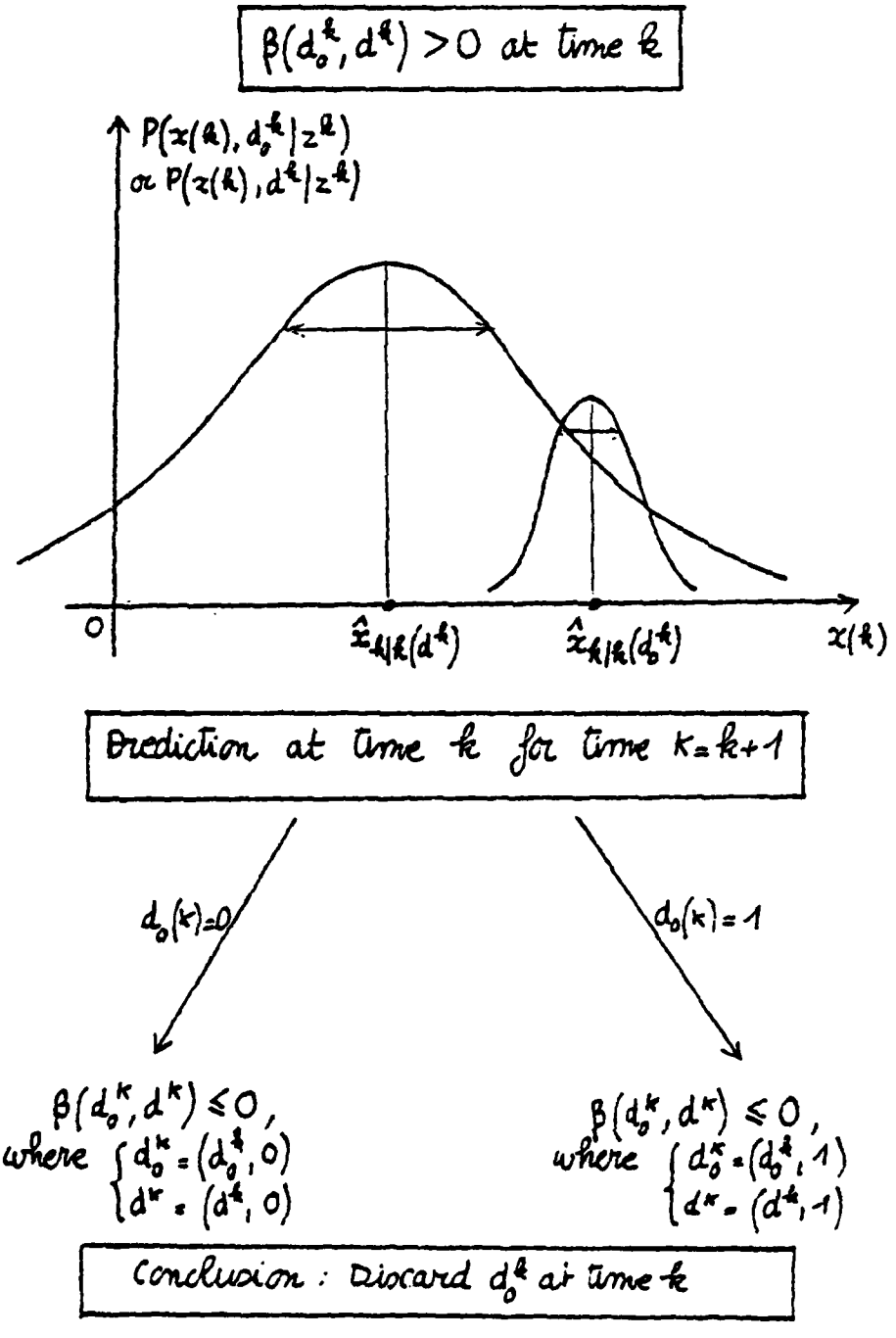
At time K, we must choose the discrete state sequence $\hat{d}^K$ among all the remaining sequences. Then, for the hybrid state estimation algorithm 2(L), an estimation of the continuous state sequence x^K can be simply done, using the Kalman Smoothing Equations of Chapter 5, conditioned on $\hat{d}^K = (\hat{d}(0),\ldots,\hat{d}(K))$; the estimate $\hat{x}^{K|K}(\hat{d}^K) = (\hat{x}_{0|K}(\hat{d}^K),\ldots,\hat{x}_{K|K}(\hat{d}^K))$ is given recursively by the following equations, where $\hat{d}(k) = p$, $\hat{d}(k+1) = q$: for $k = K-1,K-2,\ldots,0$,

$$\hat{x}_{k|K}(\hat{d}^K) = \hat{x}_{k|k}(\hat{d}^k) + L_k(\hat{d}^K)[\hat{x}_{k+1|K}(\hat{d}^K) - \hat{x}_{k+1|k}(\hat{d}^{k+1})] \quad (7.24)$$

$\hat{x}_{K|K}(\hat{d}^K)$ is the boundary condition for $k = K-1$

$$L_k(\hat{d}^K) = P_{k|k}(\hat{d}^K)F^T(p,q,k)P^{-1}_{k+1|k}(\hat{d}^{k+1}) \quad (7.25)$$

FIGURE 7.2


$\beta(d_0^k, d^k) > 0$ at time k
$P(x(k), d_0^k | z^k)$
or $P(x(k), d^k | z^k)$
0
$\hat{x}_{k|k}(d^k)$
$\hat{x}_{k|k}(d_0^k)$
$x(k)$
Prediction at time k for time $\kappa = k+1$
$d_0(\kappa) = 0$
$d_0(\kappa) = 1$
$\beta(d_0^\kappa, d^\kappa) \leq 0$,
where $\begin{cases} d_0^\kappa = (d_0^k, 0) \\ d^\kappa = (d^k, 0) \end{cases}$
$\beta(d_0^\kappa, d^\kappa) \leq 0$,
where $\begin{cases} d_0^\kappa = (d_0^k, 1) \\ d^\kappa = (d^k, 1) \end{cases}$
Conclusion : Discard d_0^k at time k

$$P_{k|K}(\hat{d}^K) = P_{k|k}(\hat{d}^k) + L_k(d^K)[P_{k+1|K}(\hat{d}^K) - P_{k+1|k}(\hat{d}^{k+1})] \quad (7.26)$$

$\hat{x}_{k|k}(\hat{d}^k)$, $\hat{x}_{k+1|k}(\hat{d}^{k+1})$, $P_{k|k}(\hat{d}^k)$, $P_{k+1|k}(\hat{d}^{k+1})$ are given by (7.2)-(7.7).

Using all the above results, we have the following algorithm for estimating the discrete state sequence d^K, according to the first objective of Chapter 6:

ALGORITHM 1(L)

STEP 0:
$$\begin{cases} \mathcal{D}^0 = d_0 \\ \hat{x}_{0|-1} = x_0 \\ P_{0|-1} = P_0 \end{cases}$$

STEP k:
$$\begin{cases} \mathcal{D}^{k-1} \\ \hat{x}_{k-1|k-1}(d^{k-1}),\ P_{k-1|k-1}(d^{k-1}) \\ P(d^{k-1}|z^{k-1}) \quad , \qquad \text{for all } d^{k-1} \in \mathcal{D}^{k-1} \end{cases}$$

RECEIVE $z(k)$

COMPUTE $\hat{x}_{k|k}(d^k)$, $P_{k|k}(d^k)$, $P(d^k|z^k)$ for all $d^k \in \mathcal{D}^{k-1} \times D$.

APPLY R1(L): for all $d_0^k \in \mathcal{D}^{k-1} \times D$, there exists $d^k \in \mathcal{D}^{k-1} \times D - \{d_0^k\}$ such that:

(i) $P(d_0^k|z^k) \underset{\text{NO}}{\overset{\text{YES}}{\leq}} P(d^k|z^k)$

YES: Continue

NO: Store d_0^k in $\mathcal{D}^k$

(ii) $P_{k|k}(d_0^k) \underset{\text{NO}}{\overset{\text{YES}}{<}} P_{k|k}(d^k)$

YES: Continue

NO: Store d_0^k in $\mathcal{D}^k$

(iii) $\beta(d_0^k, d^k) \underset{NO}{\overset{YES}{\lessgtr}} 0$

YES: Delete d_0^k

NO: Continue to AR1(L)

APPLY AR1(L): for all $d_0(k+1),\ldots,d_0(K)$, there exists d^k verifying (7.22)

YES: Delete d_0^k

NO: Store d_0^k in $\mathcal{D}^k$

$k = k+1$ and repeat until $k = K$.

STEP K: $\begin{cases} \mathcal{D}^K \\ P(d^K | z^K) \text{ for all } d^K \in \mathcal{D}^K \end{cases}$.

Find $\sup_{d^K \in \mathcal{D}^K} P(d^K | z^K)$. The corresponding state sequence is equal to $\hat{d}^K$. END.

We have also Algorithm 2(L) for estimating the hybrid state sequence $s^K = (d^K, x^K)$:

ALGORITHM 2(L)

STEP 0: $\begin{cases} \mathcal{D}^0 = d_0 \\ \hat{x}_{0|-1} = x_0 \\ P_{0|-1} = P_0 \end{cases}$

STEP k: $\begin{cases} \mathcal{D}^{k-1} \\ \hat{x}_{k-1|k-1}(d^{k-1}),\ P_{k-1|k-1}(d^{k-1}),\ P^{k-1|k-1}(d^{k-1}), \\ P(d^{k-1}|z^{k-1}) \quad \text{for all } d^{k-1} \in \mathcal{D}^{k-1} \end{cases}$

RECEIVE $z(k)$

COMPUTE: $\hat{x}_{k|k}(d^k)$, $P_{k|k}(d^k)$, $P^{k|k}(d^k)$, $P(d^k|z^k)$ for all $d^k \in \mathcal{D}^{k-1} \times D$.

APPLY R2(L): for all $d_0^k \in \mathcal{D}^{k-1} \times D$, there exists $d^k \in \mathcal{D}^{k-1} \times D - \{d_0^k\}$ such that:

(i) $P(d_0^k|z^k) \underset{\text{NO}}{\overset{\text{YES}}{\leq}} P(d^k|z^k)$

YES: Continue

NO: Store d_0^k in $\mathcal{D}^k$

(ii) $P_{k|k}(d_0^k) \underset{\text{NO}}{\overset{\text{YES}}{<}} P_{k|k}(d^k)$

YES: Continue

NO: Store d_0^k in $\mathcal{D}^k$

(iii) $\gamma(d_0^k, d^k) \underset{\text{NO}}{\overset{\text{YES}}{\leq}} 0$

YES: Delete d_0^k

NO: Continue to AR2(L)

APPLY AR2(L): for all $d_0(k+1),\ldots,d_0(K)$, there exists d^k verifying (7.23)

YES: Delete d_0^k

NO: Store d_0^k in $\mathcal{D}^k$

$k = k+1$ and repeat until $k = K$.

STEP K: $\begin{cases} \mathcal{D}^K \\ p^{K|K}(d^K),\ P(d^K|z^K) \text{ for all } d^K \in \mathcal{D}^K \\ \hat{x}_{K|K}(d^K) \end{cases}$

Find $\underset{d^K \in \mathcal{D}^K}{\text{Sup}} \frac{1}{\sqrt{|P^{K|K}(d^K)|}} \times P(d^K|z^K)$. The corresponding discrete state sequence is equal to $\hat{d}^K$. Finally, $\hat{s}^K = (\hat{d}^K, \hat{x}^{K|K}(\hat{d}^K))$ where $\hat{x}^{K|K}(\hat{d}^K) = \{\hat{x}_{k|K}(\hat{d}^K),\ k = K-1, K-2, \ldots, 0\}$ is given recursively by (7.24)-(7.25). END.

Algorithm 1(L) and 2(L) are easily implementable. We see that both objectives do not necessarily yield the same estimate of d^K, since, in the first case, we choose the most likely sequence d^K, and in the second one, we pick the sequence d^K for which $\frac{P(d^K|z^K)}{\sqrt{|P^{K|K}(d^K)|}}$ is maximum, which is not necessarily the most likely.

7.5 Summary and Conclusion

Under the Linear-Gaussian assumption, pruning rules R1(L) and R2(L) used for estimating the discrete or hybrid state sequence of a Markov process are easily implementable. Another pruning rule specific to the LG case has also been derived. These implementable pruning rules lead to the derivation of algorithms 1(L) and 2(L), which are more restricted than Algorithm 1 and 2, but are actually implementable.

A further problem would be to compare the number of sequences we

could delete optimally this way, with suboptimal algorithms that simply drop very unlikely sequences during the graph formation.

In the next chapter, we apply algorithm 1(L) to a simple failure detection problem.

CHAPTER 8

EXAMPLE: APPLICATION TO FAILURE DETECTION

The purpose of this chapter is to apply Algorithm A1(L) to a simple failure detection problem. In particular, we will look at the efficiency of our optimal pruning rule R1(L).

8.1 A Failure Detection Problem

We consider a one-dimensional dynamic system, whose continuous state is observed through a sensor subject to intermittent failures occurring at random times: in the normal case, the sensor provides state observations; in case of failure, only a burst of noise is received. The role of the failure detection system is to decide upon the time and duration of sensor failure, with a minimum probability of an error: this is an alarm task.

We apply the single hybrid Gauss-Markov model of Chapter 2 Section 2 to this problem, and pick some numerical values.

The discrete state d(k) of the system at time k can take two values:

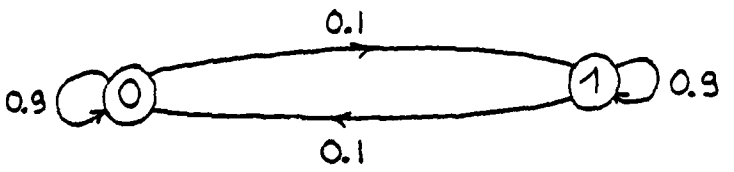
$$d(k) = \begin{cases} 0 \text{ if the sensor has not failed} \\ 1 \text{ if the sensor has failed.} \end{cases}$$

The continuous state x(k) belongs to R. We have:

- the discrete dynamics:

$$P(d(k+1) = 1 | d(k) = 0) = 0.1$$

$$P(d(k+1) = 0 | d(k) = 1) = 0.1$$



- the continuous dynamics: for $d(k) = 0,1, x(k+1) = x(k) + w(k)$

  where $w(k) \sim N(0,1)$

- the measurement equation:

  if $d(k) = 0$, then $z(k) = x(k) + v_0(k)$ where $v_0(k) \sim N(0,1)$

  if $d(k) = 1$, then $z(k) = v_1(k)$ where $v_1(k) \sim N(0,50)$

Since the task of the failure detection system is only to detect the sensor failures, we use algorithm A1(L), which estimates the discrete state sequence of our system.

8.2 Results

The system is observed during an interval [0,8]. We assume that a noise burst, resulting from a sensor failure, arises at k = 2,3,4. We study, for this case, the efficiency of pruning rule R1(L).

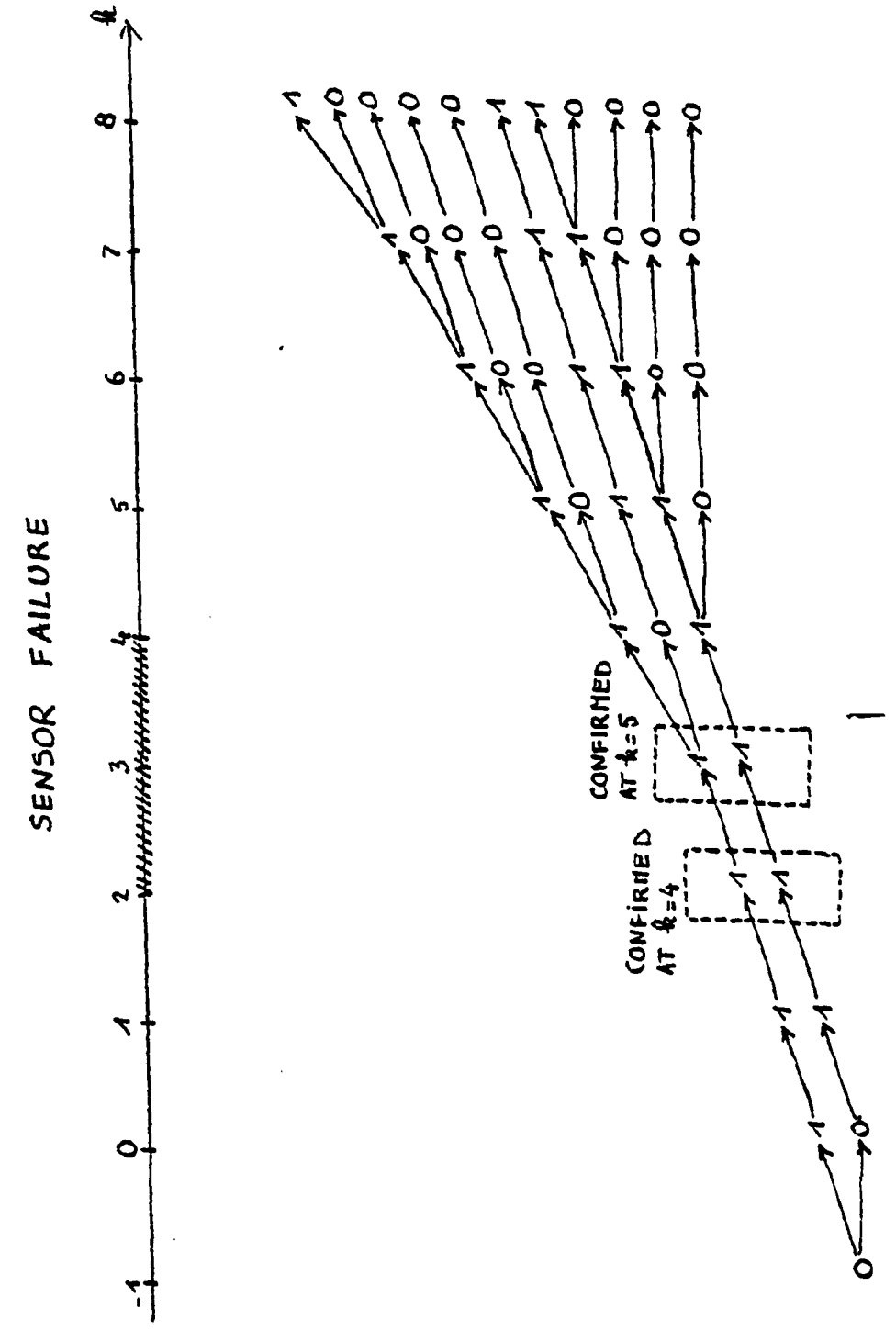
We take the following values for the sequence of observations:

k	0	1	2	3	4	5	6	7	8
z(k)	3	2.5	15	-10	10	0	3	2.5	2.5

The application of pruning rule R1(L) in the graph of the possible discrete state sequences yields the diagram shown in Fig. 8.1

We see that the algorithm is working pretty well. At time 8, we are left with only 11 different discrete state sequences, using R1(L)

FIGURE 8.1


SENSOR FAILURE
k
-1
0
1
2
3
4
5
6
7
8
CONFIRMED AT k=5
CONFIRMED AT k=4

(the maximum number of possible sequences is 512). Furthermore, the algorithm confirmed at time 4 and 5, that a failure has occurred at time 2 and 3 respecitvely. Therefore, it responds quickly to the sensor failure.

So, the application of A1(L) to the multiple hypothesis filter-detectors, mentioned in Chapter 3, should lead to an optimal procedure for detecting a failure, that keeps the amount of computation at a reasonable level.

This example highlights the efficienty of our pruning rule in a very simple case. Of course, more complete simulations would have to be run, to estimate the real capability of our algorithm.

PART III

THE MULTIPLE SYSTEM CASE

CHAPTER 9

STATE ESTIMATION OF A MULTIPLE HYBRID MARKOV PROCESS

In part II, the state estimation of a single hybrid Markov process has been studied. We now consider the more complex problem of estimating the state sequence of multiple hybrid Markov processes in parallel.

9.1 Problem Statement

The multiple system model considered here is the one described in Chapter 2 Section 3.

As for the single system problem of Chapter 6, there are two possible objectives for a state estimation of a multiple hybrid system, according to the application. One possible objective is the MAP estimation of the total hybrid state sequence $s^K = (d^K, x^K)$ of the multiple system. In other cases, e.g., in some multi-target tracking, we could be more interested in the number of targets in a certain area, and thus would use an MAP estimation of the discrete state sequence d^K of the multiple systems. Then, conditioned on the estimate of d^K, we can look for the MAP estimate of the continuous state sequence x^K.

Because of the Markov and memoryless properties (2.9)-(2.11), we can apply directly algorithm 1 or 2 of Chapter 6 to the multiple system, which is the cartesian product of the individual systems. However, such an algorithm would not take advantage of a possible decomposition of the problem, due to the relative independence among the subsystems. If we could find, in both cases, the optimal estimates by running in parallel

p independent algorithms 1 or 2, one for each subsystem, then the amount of computation would be greatly reduced: the total number of possible sequences in the corresponding p independent graphs is $p \times N^K$, which is much smaller than the number of possible sequences in the corresponding graph for the multiple system, equal to $(N^p)^K$.

9.2 Decomposition of the Estimation Problem

The purpose of this section is to find the assumptions that would make the general objective of the estimation problem equivalent to a set of local independent objectives, in order to get a decomposition of our algorithm.

As mentioned in Chapter 2 Section 3, the MAP hybrid state estimate of s^K given by s^K for which $P(s^K|Y^K)$, or equivalently $P(Y^K|s^K) \times P(s^K)$ is maximum. Using the Markov and memoryless properties, we have:

$$P(Y^K|s^K) \times P(s^K) = \prod_{k=0}^{K} P(Y(k)|s(k)) \times P(s(k)|s(k-1)) \qquad (9.1)$$

Similarly, for the discrete state estimation of the multiple system, the objective is to find $\hat{d}^K = (\hat{d}(0),\ldots\hat{d}(K))$ for which:

$$P(Y^K|d^K) \times P(d^K) = \prod_{k=0}^{K} P(Y(k)|d^k,Y^{k-1}) \times P(d(k)|d(k-1)) \qquad (9.2)$$

is maximum.

We recall the independent dynamics and independent measurement generation processes assumptions that we made in Chapter 2:

Assumption (A.1): The dynamics of the subsystems are assumed to be independent, i.e., for all $k = 0,1,\ldots,K$,

$$P(s(k)|s(k-1)) = \prod_{i=0}^{p} P(s_i(k)|s_i(k-1)) \tag{9.3}$$

Assumption (A.2): The sets of measurements generated by the subsystems and the sets of false alarms are assumed to be statistically independent, i.e., for all k = 0,1,...,K ,

$$P(Y_1(k),\ldots,Y_p(k),Y_f(k)|s(k)) = \prod_{i=1}^{p} P(Y_i(k)|s_i(k)) \times P(Y_f(k)) \tag{9.4}$$

This implies, that for all k = 0,1,...,K,

$$P(Y_1(k),\ldots,Y_p(k),Y_f(k)|d^k,Y_1^{k-1},\ldots,Y_p^{k-1},Y_f^{k-1}) =$$

$$\prod_{i=1}^{p} P(Y_i(k)|d_i^k,Y_i^{k-1}) \times P(Y_f(k)) \tag{9.5}$$

At this point, if the partition of Y(k) into $\{Y_1(k),\ldots,Y_p(k), Y_f(k)\}$ were a priori known, then, under assumptions (A.1)-(A.2), our estimation problems would be equivalent to find for each i = 1,2,...,p, the sequence s_i^K for which $\prod_{k=0}^{K} P(Y_i(k)|s_i(k)) \times P(s_i(k)|s_i(k-1))$ is maximum, or under the second objective, the sequence d_i^K for which $\prod_{k=0}^{K} P(Y_i(k)|d_i^k,Y_i^{k-1}) \times P(d_i(k)|d_i(k-1))$ is maximum. The estimation problem would be solved by p independent subproblems.

Unfortunately, as we emphasized in Chapter 2, in general this partition is unknown. We can only make a hypothesis for the partition of Y(k) at time k, that is called a data hypothesis at time k, denoted by A(k). We saw that the probability distribution for the set of measurements Y(k) is equal to:

$$P(Y(k)|s(k)) = \sum_{\{A(k)\}} \prod_{i=1}^{p} P(Y_i(A(K))|s_i(k)) \times P(Y_f(A(k)) \times P(A(k)|d(k)) \quad (9.6)$$

and

$$P(Y(k)|d^k,Y^{k-1}) = \sum_{\{A^K\}} \prod_{i=1}^{p} P(Y_i(A(k))|d_i^k,Y_i^{k-1}(A(k))) \times P(Y_f(A(k)) \times P(A^k|d^k) \quad , \quad (9.7)$$

where $Y_1(A(k)),\ldots,Y_p(A(k))$, $Y_f(A(k))$ denotes the hypothesized partition of $Y(k)$ under $A(k)$.

Thus, even though the system dynamics and measurement processes are independent, the data association uncertainty introduces a dependancy among the subsystems for the estimation problem. If we cannot reduce the uncertainty in the data hypotheses by some further assumptions, then no decomposition is possible.

As mentioned above, an a priori knowledge of the partition $Y_1(k),\ldots,Y_p(k),Y_f(k)$ would lead to this decomposition, since it would imply that only one data association is possible and (9.6)-(9.7) would reduce to (9.4)-(9.5) respectively.

Some weaker assumptions can also lead to the decomposition. For this purpose, we define the following regions of the measurement space Z.

Definition: A measurement region $R_i(k)$ is defined as the set of measurements that can a priori come from subsystem i at time k.

In some digital estimation problems, it can correspond to the possible discrete values of the measurement emitted by a subsystem. In

multitarget tracking problems, as we shall see in Chapter 11, it corresponds to a region, outside which no measurement can arise from the target.

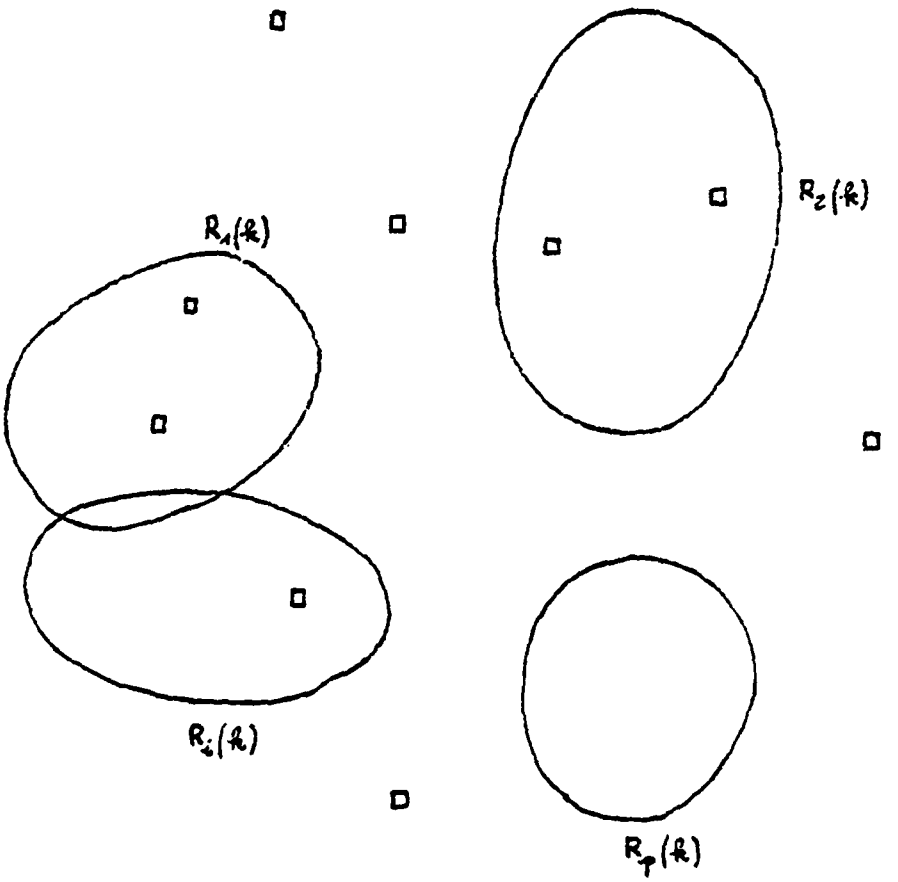
If at each time k, the sets $Y(k) \cap R_i(k)$ are mutually exclusive, then the data association uncertainty is greatly reduced. A partial partition of the set of observations Y(k) is possible: the measurements located outside any regions $R_i(k)$ can definitely be assigned to the set of false alarms. Each measurement inside $R_i(k)$, however, can be assigned either to subsystem i or to the set of false alarms. If the detection and false alarm processes are such that the numbers of detections, missed detections, and false alarms in a given region are statistically independent of the numbers in any disjoint region, then nothing correlates anymore the different subsystems and the decomposition of our estimation problem should follow. This assumption is weaker than the one that would suppose an a priori knowledge of the sets $Y_1(k),\ldots,Y_p(k)$, $Y_f(k)$, since there remains a data association uncertainty inside each region $R_i(k)$, between actual observations and false alarms. (See Fig. 9.1)

This independence assumption can be stated formally as follows. If at time k, the sets $Y(k) \cap R_i(k)$ are mutually exclusive, then we partition Y(k) in:

$$Y(k) = \{Y(k) \cap R_1(k),\ldots,Y(k) \cap R_p(k),\ Y(k) \cap [Z - \bigcup_{i=1}^{p} R_i(k)]\}.$$

A measurement in $Y(k) \cap [Z - \bigcup_{i=1}^{p} R_i(k)]$ is associated to the set of false alarms $Y_f(k)$, since it lies outside any region $R_i(k)$. In this case, the data hypothesis A(k) takes the form: $A(k) = \{A_1(k),\ldots,A_p(k), A_f(k)\}$, where $A_i(k)$ is the corresponding data hypothesis from $d_i(k)$

FIGURE 9.1



$R_i(k)$: measurement region at time k

□ : measurements at time k

into $Y(k) \cap R_i(k)$, and $A_f(k)$, and $A_f(k)$ is the association of the measurements in $Y(k) \cap (Z - \bigcup_{i=1}^{p} R_i(k))$ to the set of false alarms $Y_f(k)$. We make the following assumption:

Assumption (A.3): If, at time k, the sets $Y(k) \cap R_i(k)$ are mutually exclusive, then we assume that all the data hypotheses $A_1(k),\ldots,A_p(k)$, $A_f(k)$ are statistically independent, i.e., $P(A(k)|d(k)) = \prod_{i=1}^{p} P(A_i(k)|d_i(k)) \times P(A_f(k))$.

In a multitarget tracking application, if the detectability of each target is independent of the others, and if the process generating a false alarm is independent of the other false alarms, then assumption (A.3) is satisfied. But, as it is often the case, if the number of false alarms in the area is assumed to follow a Poisson distribution then Assumption (A.3) would not hold; such a model for the false alarms would prevent any decomposition of our problem.

Thus, if the regions $Y(k) \cap R_i(k)$ are mutually exclusive, then the possible data hypotheses take the form: $A(k) = \{A_1(k),\ldots,A_p(k), A_f(k)\}$ defined above. In this case, we have for the hybrid state estimation problem under (A.2)-(A.3),

$$P(Y(k)|s(k)) = \sum_{\{A(k)\}} P(Y(k)|s(k),A(k)) \times P(A(k)|d(k))$$

$$= \sum_{\{A_1(k)\},\ldots,\{A_p(k)\}} \prod_{i=1}^{p} P(Y(k) \cap R_i(k)|s_i(k),A_i(k)) \times$$

$$P(A_i(k)|d_i(k)) \times P(Z_f(k)) \times P(A_f(k)) = P(Z_f(k)) \times$$

$$P(A_f(k)) \times \prod_{i=1}^{p} \sum_{\{A_i(k)\}} P(Y(k) \cap R_i(k)|s_i(k),A_i(k)) \times$$

$$P(A_i(k)|d_i(k)) = P(Z_f(k) \times P(A_f(k)) \times \prod_{i=1}^{p} P(Y(k) \cap R_i(k)|s_i(k)) \quad ,$$

where $Z_f(k)$ denotes $Y(k) \cap [Z - \bigcup_{i=1}^{p} R_i(k)]$.

So, using (A.1), the problem of finding $\hat{s}^K$ for which $P(s^K|Y^K)$ is maximum, is strictly equivalent to the p following independent subproblems: for all i = 1,2,...,p, find $\hat{s}_i^k$ for which:

$$\prod_{k=0}^{K} P(Y(k) \cap R_i(k)|s_i(k)) \times P(s_i(k)|s_i(k-1)) \text{ is maximum; and}$$

$\hat{s}^K = (\hat{s}_1^K,\ldots,\hat{s}_p^K)$.

Similarly, for the discrete state estimation problem, we can show under the same assumptions that:

$$P(Y(k)|Y^{k-1},d^k) = P(Z_f(k)) \times P(A_f(k)) \times \prod_{i=1}^{p} P(Y(k) \cap R_i(k)|$$

$$Y_i^{k-1},d_i^k).$$

Therefore, the problem of finding $\hat{d}^K$ for which $P(d^K|Y^K)$ is maximum is equivalent to the following independent subproblems: for all i = 1,2,..., p, find $\hat{d}_i^K$ for which:

$$\prod_{k=0}^{K} P(Y(k) \cap R_i(k)|Y_i^{k-1},d_i^k) \times P(d_i(k)|d_i(k-1)) \text{ is maximum;}$$

and $\hat{d}^K = (\hat{d}_1^K,\ldots,\hat{d}_p^K)$.

We summarize the above results as follows:

Main Result: Under assumptions (A.1)-(A.2)-(A.3), if, at each time k = 0,1,...,K, the sets $Y(k) \cap R_i(k)$ are mutually exclusive, then the multiple hybrid and discrete estimation problems are equivalent to a set of p independent subproblems, one for each subsystem.

Thus, instead of running one algorithm for the multiple system, we would run p algorithms in parallel, similar to the ones of Chapter 6. This would lead to an enormous simplification, since, before any deletions, the total number of possible state sequences at time K would be reduced from $(N^p)^K$ to $p \times N^K$.

If some of the sets $Y(k) \cap R_i(k)$ do overlap during the period [0,K], then a partial decomposition of the problem is still possible, by running, as in Reid's algorithm [9], one algorithm for the cluster of the corresponding subsystems, and independent algorithms for the remaining subsystems.

A further problem is to know, if, after the overlapping of some sets, we can still recover the decomposition property of our algorithm, when they separate again. This is the subject of the following section.

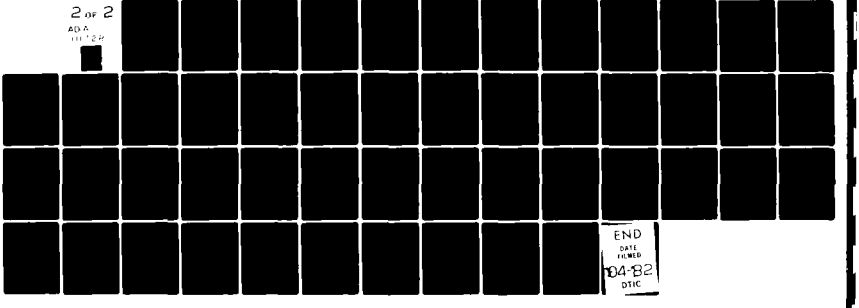
9.3 Actual Implementation of an Algorithm

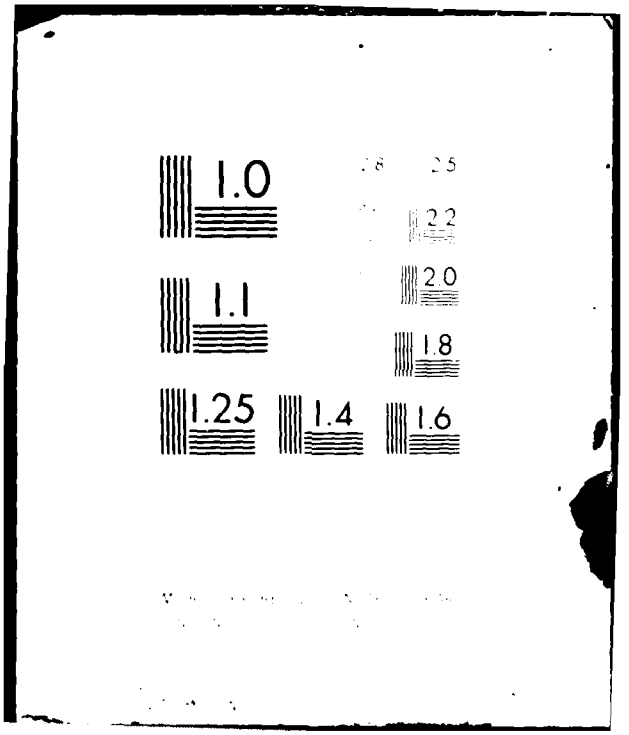
An algorithm for estimating the hybrid or discrete state sequence of a multiple Markov process must take advantage of any decomposition opportunity for reducing the amount of computation.

The multiple system is observed during an interval [0,K]. As long as no overlap of any sets $Y(k) \cap R_i(k)$ has ever occurred, we can run p algorithms in parallel. If, at a given time k, some overlapping takes place, then we can still revert to independent algorithms, if, using the pruning rules, we find a single remaining survivor through the interval of overlap, since this confirms the multiple system state during this interval. Otherwise, the resulting data association uncertainty introduces a correlation among our subsystems, that is going

AD-A111 728 MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMA--ETC F/G 12/1
STATE ESTIMATION OF A HYBRID MARKOV PROCESS WITH APPLICATION TO--ETC(U)
JAN 82 F E BRUNEAU N00014-77-C-0532
UNCLASSIFIED LIDS-TH-1172 NL

2 OF 2
AD A
111728




1.0
1.1
1.25
1.4
1.6
2.2
2.0
1.8

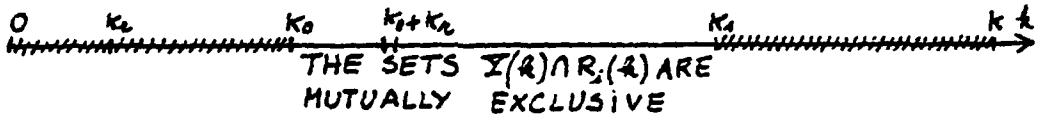
to last until time K. So, after that time k, one algorithm has to be run for the overall system.

In many cases, the number of remaining state sequences at time K is still going to be well beyond our storage capabilities. Then, the algorithms are truncated to some manageable length K_r, much smaller than K, in order to keep the number of hypothesized sequences at a reasonable level. In these K_r-scan algorithms, for each time $k = K_r, K_r+1,\ldots,K$, we compute recursively the discrete or hybrid state estimate at time $k-K_r$ as follows: having decided upon the state estimates $\hat{d}[k-K_r-1]$ or $\hat{s}[k-K_r-1]$ at time k-1, we determine the most likely discrete or hybrid state sequence from $k-K_r-1$ to k, using the pruning rules of Algorithm 1 or 2 of Chapter 6. Then, the corresponding discrete or hybrid state $\hat{d}[k-K_r]$ or $\hat{s}[k-K_r]$ on this sequence is chosen as the state estimate at time $k-K_r$. Since at time k-1, the state at time $k-K_r-1$ has been confirmed, we can run p independent algorithms, if, from to $k-K_r$ to k, the observation sets defined in Section 2 are mutually exclusive. Thus, this guarantees a decomposition, even though there has been some overlap before time $k-K_r$. (See Fig. 9.2).

This kind of situation often arises in multitarget tracking. When the surveillance system has just been initialized, we have a very poor picture of the area with much uncertainty in the location of the different targets. After a while, the target state estimates become more accurate and, if they are largely spaced, the regions $Y(k) \cap R_i(k)$ become mutually exclusive. Then, some crossings can occur which introduce again a correlation between the different target state estimations for a non-zero, but finite, duration.

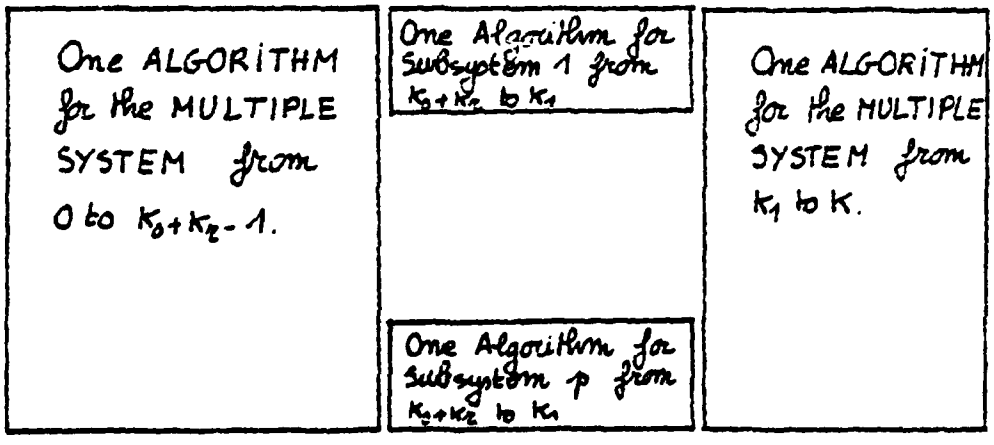
FIGURE 9.2

TIME



THE SETS $\Sigma(t) \cap R_s(t)$ ARE MUTUALLY EXCLUSIVE

STRUCTURE



9.4 Summary

The computational complexity of an algorithm for estimating the discrete or hybrid state sequence of a multiple Markov process led us to consider a decomposition of this problem into several independent subproblems. In order to get this decomposition, some model assumptions, as independent dynamics, independent measurements, independent detection and false alarms, had to be made, and a condition on the sets $Y(k) \cap R_i(k)$ had to be met. For the application to multitarget tracking in Chapter 11, the actual computation of these regions $R_i(k)$ will have to be done. This will show that our multitarget tracking problem under the above assumptions can be equivalent to p independent single target tracking problems.

We next consider an additional Linear-Gaussian structure on the multiple Markov process.

CHAPTER 10

STATE ESTIMATION OF A MULTIPLE GAUSS-MARKOV PROCESS

This chapter is the final stage before the application to the multi-target tracking problem. The purpose is to get an actually implementable algorithm, that connects the results of the last chapter on the decomposition of the estimation problem, to the results of Chapter 7 on the estimation of a single Gauss-Markov process.

10.1 Problem Statement

The system considered here is the one of Chapter 2 Section 3, under the assumptions (A.1)-(A.2)-(A.3) of the last chapter, plus an additional Linear-Gaussian assumption.

As in Chapter 7, an algorithm for estimating the discrete or hybrid state sequence of the multiple system will use some pruning rules based on the distributions $P(x(k),d^k|Y^k)$ and $\overline{P}(x(k),d^k|Y^k)$, where now $x(k) = \{x_1(k),\ldots,x_p(k)\}$ and $d^k = \{d_1^k,\ldots,d_p^k\}$. However, under the Linear-Gaussian assumption, the expressions for these distributions arenot as simple as for the single Gauss-Markov process of Chapter 7. We have:

$$P(x(k),d^k|Y^k) = P(d^k|Y^k) \sum_{\{A^k\}} P(x(k)|d^k,Y^k,A^k) \times P(A^k|d^k) \quad (10.1)$$

$P(x(k)|d^k,Y^k,A^k)$ is the conditional density for the multiple state $x(k)$ on a discrete state trajectory d^k and a data hypothesis sequence A^k up to time k. The dynamics' independence implies that on a given d^k and A^k:

$$P(x(k)|d^k,Y^k,A^k) = \prod_{i=1}^{p} P(x_i(k)|d^k,Y^k,A^k) \tag{10.2}$$

Each distribution $P(x_i(k)|d^k,Y^k,A^k)$ is a Gaussian density, whose mean $\hat{x}^i_{k|k}(d^k,A^k)$ and covariance $P^i_{k|k}(d^k,A^k)$ depend on both d^k and A^k, and can be determined recursively from the Kalman Filtering Equations (7.2)-(7.7), where, at each time k, the state estimate is updated with the measurement, if any, given by A(k).

Therefore, $P(x(k)|d^k,Y^k,A^k)$ can be written as $N[x(k) - \hat{x}_{k|k}(d^k,A^k), P_{k|k}(d^k,A^k)]$ where:

$$\hat{x}_{k|k}(d^k,A^k) = \begin{bmatrix} \hat{x}^1_{k|k}(d^k,A^k) \\ \vdots \\ \hat{x}^p_{k|k}(d^k,A^k) \end{bmatrix} \text{ and } P_{k|k}(d^k,A^k) =$$

$$\begin{bmatrix} P^1_{k|k}(d^k,A^k) & & 0 \\ & \ddots & \\ 0 & & P^p_{k|k}(d^k,A^k) \end{bmatrix}$$

and $P(x(k),d^k|Y^k) = P(d^k|Y^k) \sum_{\{A^k\}} P(A^k|d^k) \times N[x(k) - \hat{x}_{k|k}(d^k,A^k), P_{k|k}(d^k,A^k)]$. (10.3)

Similarly, we have:

$$\bar{P}(x(k),d^k|Y^k) = P(d^k|Y^k) \sum_{\{A^k\}} P(A^k|d^k) \sqrt{\frac{(2\pi)^{nk}|P_{k|k}(d^k,A^k)|}{(2\pi)^{pnk}|P^{k|k}(d^k,A^k)|}} \times$$

$$N[x(k) - \hat{x}_{k|k}(d^k,A^k), P_{k|k}(d^k,A^k)] \tag{10.4}$$

where $P^{k|k}(d^k,A^k)$ is the covariance of the state sequence x^k conditioned on d^k,A^k.

So, for both algorithms the pruning rules in the graph would be based on a functional inequality between two linear sums of Gaussian distributions. Unfortunately, no simple test can be found that is equivalent to such an inequality. However, some pruning rules based on $P(x(k),d^k,A^k|Y^k)$ or $\tilde{P}(x(k),d^k,A^k|Y^k)$ would be easily implementable using the Lemma of Chapter 7. Thus, instead of estimating the hybrid or discrete state sequence, we can think of estimating the pair (d^K,A^K) or (s^K,A^K). The resulting $\hat{d}^K$ or $\hat{s}^K$ are taken as our state sequence estimates.

10.2 Suboptimal Algorithm for Estimating the State of a Multiple Gauss-Markov Process

In this section, we develop an algorithm for estimating the pair (d^K,A^K) or (s^K,A^K) of our multiple system. The derivation is very close to the one of Algorithm 1 or 2 of Chapter 7.

We define the pair $(d(k),A(k))$ as being a hypothesis at time k on the multiple system. A hypothesis tree can be associated with our system. In this tree, each node corresponds to a distinct hypothesis $h(k)$ at time k and each branch represents a transition to some new hypothesis at the next instant of time. (See Fig. 10.1).

According to the last section, the distributions $P(x(k),h^k|Y^k)$ and $P(x(k),h^k|Y^k)$ are equal to:

$$P(x(k),h^k|Y^k) = P(h^k|Y^k) \times N[x(k) - \hat{x}_{k|k}(h^k),P_{k|k}(h^k)], \quad (10.5)$$

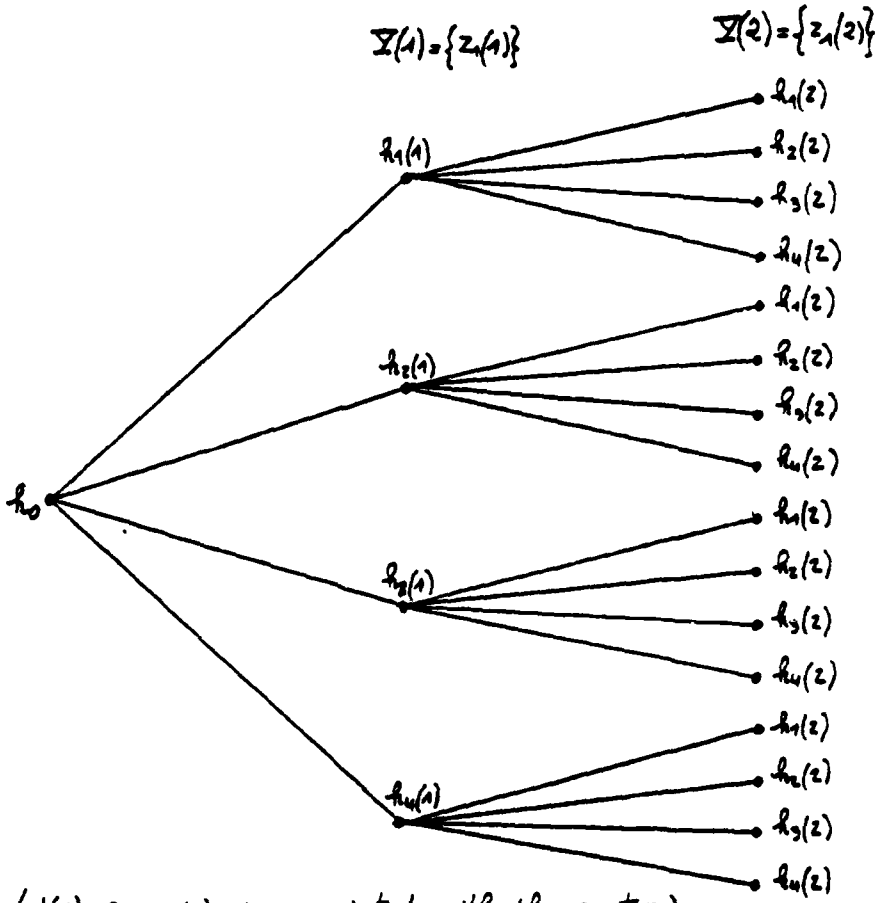
$$P(x(k),h^k|Y^k) = P(h^k|Y^k) \times \sqrt{\frac{(2\pi)^{np}|P_{k|k}(h^k)|}{(2\pi)^{npk}|P^{k|k}(h^k)|}}$$

$$\times N[x(k) - \hat{x}_{k|k}(h^k),P_{k|k}(h^k)] \quad (10.6)$$

FIGURE 10.1

HYPOTHESIS TREE

Assume $p=1$, $d(k) \in \{0,1\}$ and $h(0) = h_0$



$h_1(1) = (d(1)=0,\ z_1(1)$ is associated with the system)

$h_2(1) = (d(1)=0,$ the system is not detected and $z_1(1)$ is a false alarm)

$h_3(1) = (d(1)=1,\ z_1(1)$ is associated with the system)

$\vdots$

The a posteriori probability $P(h^k|Y^k)$ is equal to $P(h^k,Y^k)/P(Y^k)$ where $P(Y^k)$ is a normalization term; and $P(h^k,Y^k)$ can be determined recursively from:

$$P(h^k,Y^k) = P(A(k)|d(k)) \times P(d(k)|d(k-1)) \times P(Y(k)|h^k,Y^{k-1}) \times P(h^{k-1},Y^{k-1}) \tag{10.7}$$

$P(A(k)|d(k))$ and $P(d(k)|d(k-1))$ are given by the model assumptions. $P(Y(k)|h^k,Y^{k-1})$ is the innovation density on h^k for the multiple system. We have, using Chapter 7:

$$P(Y(k)|h^k,Y^{k-1}) = \prod_{i=1}^{p} P(z_i(k)|d^k,h^{k-1},Y^{k-1}) \times P(Y_f(A(k))$$

$$= P(Y_f(A(k)) \times \prod_{i=1}^{p} N[z_i(k) - G_i(d_i(k),k)\hat{x}^i_{k|k}(h^k), B_i(h^k)] \tag{10.8}$$

where $$B_i(h^k) = H_i(d_i(k),k)P^i_{k|k-1}(h^k)H_i^T(d_i(k),k) + R_i(d_i(k),k) \tag{10.9}$$

An algorithm for estimating the multiple sequence h^K or (h^K,x^K) would use, as for Algorithm 1 or 2 of Chapter 7, some pruning rules in the hypothesis tree based on $P(x(k),h^k|Y^k)$ or $\tilde{P}(x(k),h^k|Y^k)$ which are Gaussian distributions. To get a simple implementation of these pruning rules, we can now apply the Lemma of Chapter 7.

In the following, we only derive the pruning rules and the corresponding algorithm for estimating the discrete state sequence of the multiple system. The derivation for the hybrid state estimation algorithm is simply obtained by formally replacing $P_{k|k}(h^k)$ by $p^{k|k}(h^k)$ and β by γ.

We get immediately the following pruning rule MR1(L), that compares two sequences h_1^k and h_2^k terminating at the same discrete state d(k) at time k in the hypothesis tree:

Pruning Rule MR1(L): Let h_1^k and h_2^k be two hypothesis sequences such that $d_1(k) = d_2(k)$. Let

$$\beta(h_1^k,h_2^k) = \sum_{i=1}^{p} (\hat{x}^i_{k|k}(h_2^k) - \hat{x}^i_{k|k}(h_1^k))^T (P^i_{k|k}(h_2^k) - P^i_{k|k}(h_1^k))^{-1}$$

$$(\hat{x}^i_{k|k}(h_2^k) - \hat{x}^i_{k|k}(h_1^k)) + 2 \ln \frac{P(h_1^k|Y^k)}{P(h_2^k|Y^k)} +$$

$$\sum_{i=1}^{p} \ln \frac{|P^i_{k|k}(h_2^k)|}{|P^i_{k|k}(h_1^k)|} ,$$

where $\hat{x}^i_{k|k}(h_2^k), P^i_{k|k}(h_2^k)$, $\hat{x}^i_{k|k}(h_1^k), P^i_{k|k}(h_1^k)$ are the estimates and covariances of subsystem i on h_1^k and h_2^k. If, for all i=1,2,...,p, $P^i_{k|k}(h_1^k) < P^i_{k|k}(h_2^k)$ and $\beta(h_1^k,h_2^k) \leq 0$, then h_1^k can be immediately discarded at time k, without increasing the probability of an error in estimating the sequence h^K.

Proof: The proof comes directly from the Lemma, the expression for $P(x(k),h^k|Y^k)$ and the fact that $P(A(k)|d(k))$ depends only on the discrete state at time k.

As in Chapter 7, there exists a recursive relation on the coefficient β, that is independent of the measurements. This relation leads also to another pruning rule as follows:

Pruning Rule AMR1(L): Let h_0^k be an hypothesis sequence up to time k. If for all $h_0(k+1),\ldots,h_0(K)$, there exists a sequence $h^K = (h^k, h_0(k+1),\ldots, h_0(K))$, where the corresponding d(k) is equal to $d_0(k)$, such that:

$$\beta(h_0^k,h^k) \leq \sum_{i=1}^{p} \sum_{j=k+1}^{K} \ell n \frac{|B_i(h_0^j)|}{|B_i(h^j)|} + \sum_{i=1}^{p} \ell n \frac{|P^i_{k|k}(h^k)|}{|P^i_{k|k}(h_0^k)|} + \sum_{i=1}^{p} \ell n \frac{|P^i_{K|K}(h_0^K)|}{|P^i_{K|K}(h^K)|} ,$$

then h_0^k can be immediately deleted at time k, without increasing the probability of an error in estimating (d^K, A^K).

Using MR1(L) and AMR1(L), an algorithm, similar to Algorithm 1 of Chapter 7, can be derived for estimating the sequence (d^K, A^K) of the multiple system.

However, we must also use the decomposition possiblities, introduced in the last chapter, that can lead to a substantial reduction in the amount of computation. If, for all k = 0,1,...,K, the sets Y(k) $R_i(k)$ defined in the last chapter are mutually exclusive, then we can solve the estimation problem by running p independent algorithms. For each algorithm i, a hypothesis tree for the corresponding subsystem i is generated. Then, some pruning rules based on $P(x_i(k), h_i^k | Y^k)$ are used to delete at time k some hypothesis sequences h_i^k during the formation of tree i. Using the same arguments as in Chapter 7, these pruning rules can be stated as follows:

Pruning Rule R1(L) in tree i: Let $h^k_{1,i}$ and $h^k_{2,i}$ be two hypothesis sequences up to time k in tree i such that the corresponding $d_{1,i}(k)$ is equal to $d_{2,i}(k)$. Let

$$\beta(h_{1,i}^k, h_{2,i}^k) = (\hat{x}_{k|k}^i(h_{1,i}^k) - \hat{x}_{k|k}^i(h_{2,i}^k))^T (P_{k|k}^i(h_2^k) - P_{k|k}^i(h_1^k))^{-1} (\hat{x}_{k|k}^i(h_{1,i}^k) - \hat{x}_{k|k}^i(h_{2,i}^k)) + 2 \ln \frac{P(h_{1,i}^k | Y^k)}{P(h_{2,i}^k | Y^k)} + \ln \frac{|P_{k|k}(h_{1,i}^k)|}{|P_{k|k}(h_{2,i}^k)|} .$$

If $P_{k|k}^i(h_{1,i}^k) < P_{k|k}^i(h_{2,i}^k)$ and $\beta(h_{1,i}^k, h_{2,i}^k) \leq 0$, then $h_{1,i}^k$ can be immediately discarded at time k, without increasing the probability of an error in estimating (d_i^K, A_i^K) in tree i.

We have also the additional pruning rule:

Pruning Rule AR1(L) in tree i: Let $h_{0,i}^k$ be a hypothesis sequence in tree i up to time k. If, for all $h_{0,i}(k+1), \ldots, h_{0,i}(K)$, in tree i, there exists $h_i^K = (h_i^k, h_{0,i}(k+1), \ldots, h_{0,i}(K))$, where the corresponding $d_i(k)$ is equal to $d_{0,i}(k)$, such that $P_{k|k}^i(h_{0,i}^k) < P_{k|k}^i(h_i^k)$ and

$$\beta(h_{0,i}^k, h_i^j) \leq \sum_{j=k+1}^{K} \ln \frac{|B_i(h_{0,i}^j)|}{|B_i(h_i^j)|} + \ln \frac{|P_{k|k}^i(h_i^k)|}{|P_{k|k}^i(h_{0,i}^k)|} + \ln \frac{|P_{K|K}^i(h_{0,i}^K)|}{|P_{K|K}^i(h_i^K)|} ,$$

then $h_{0,i}^k$ can be immediately discarded at time k, without increasing the probability of an error in estimating (d_i^K, A_i^k).

If the independence among the algorithms can be preserved until time K, then the estimate $\hat{h}^K$ is simply given by $(\hat{h}_1^K, \ldots, \hat{h}_p^K)$ resulting from the subalgorithms.

If, at a given time, some overlaps occur, then we have to go back to a combined algorithm using MR1(L), AMR1(L).

The algorithm MA1(L), for estimating the pair (d^K, A^K) of the multiple system, uses the combination of the local algorithms, whenever it is possible, and the combined algorithm. We denote by $\mathcal{H}_i^k$ and $\mathcal{H}^k$ the remaining hypothesis sequences at time k in tree i and in the global tree respectively. The algorithm can be stated formally as follows:

ALGORITHM MA1(L)

STEP 0: $\begin{cases} \mathcal{H}^0 = h_0 \\ \hat{x}^i_{0|-1}(h_0) = x^i_0 \ , \ P^i_{0|-1}(h_0) = P^i_0, \ R_i(0) = R^0_i, \text{ for all} \end{cases}$

$i = 1,2,\ldots,p$.

Test: $\forall\ (i,j) \in [1,p]^2,\ Y(0) \cap R^0_i \cap R^0_j = \emptyset$

YES: Go to SUBALGORITHM SA1(L)

NO: CONTINUE

STEP k: $\begin{cases} \mathcal{H}^{k-1} \\ P(h^{k-1}|Y^{k-1}), \hat{x}^i_{k-1|k-1}(h^{k-1}), \ P^i_{k-1|k-1}(h^{k-1}) \end{cases}$

for all $i = 1,2,\ldots,p$,

for all $h^{k-1} \in \mathcal{H}^{k-1}$

RECEIVE Y(k)

COMPUTE for all $h^k \in \mathcal{H}^{k-1} \times$ all the possible multiple hypotheses at time k, $P(h^k|Y^k)$, $\hat{x}^i_{k|k}(h^k)$, $P^i_{k|k}(h^k)$.

APPLY MR1(L)

APPLY AMR1(L)

STORE the remaining sequences in $\mathcal{H}^k$

k = k+1 and repeat until k = K

STEP K: $\left\{ \mathcal{H}^K \right.$

$\left. P(h^K|Y^K) \text{ for all } h^K \in \mathcal{H}^K \right.$.

Find $\underset{h^K \in \mathcal{H}^K}{\text{Sup}} \; P(h^K|Y^K)$. The corresponding sequence is equal to $\hat{h}^K$. STOP.

SUBALGORITHM SA1(L):

STEP k: for all i = 1,...,p, $\left\{ \mathcal{H}_i^{k-1} \right.$

$\left. P(h_i^{k-1}|Y_i^{k-1}),\ \hat{x}^i_{k-1|k-1}(h_i^{k-1}),\ P^i_{k-1|k-1}(h_i^{k-1}), \text{ for all } h_i^{k-1} \in \mathcal{H}_i^{k-1} \right.$

COMPUTE for all i = 1,...,p, $R_i(k)$

(A method for computing $R_i(k)$ will be derived for the multitarget application) .

TEST: for all $(i,j) \in [1,p]^2$, $Y(k) \cap R_i(k) \cap R_j(k) = \emptyset$

YES: Continue

NO: Go to Step k of Algorithm MA1(L).

RECEIVE Y(k)

COMPUTE for all i = 1,2,...,p, for all $h_i^k \in \mathcal{H}_i^{k-1}$ × all the possible hypotheses in tree i at time k, $\hat{x}^i_{k|k}(h_i^k)$, $P^i_{k|k}(h_i^k)$, $P(h_i^k|Y_i^k)$

APPLY R1(L) in tree i, for all i = 1,...,p

APPLY AR1(L) in tree i, for all i = 1,...,p

STORE the remaining sequences in $\mathcal{H}_i^k$

k = k+1 and repeat until k = K

STEP K: For all $i = 1,\ldots,p$ compute $\mathcal{H}_i^K$

$P(h_i^K | Y_i^K)$ for all $h_i^K \in \mathcal{H}_i^K$.

Find $\sup_{h_i^K \in \mathcal{H}_i^K} P(h_i^K | Y_i^K)$. The corresponding sequence is our estimate $\hat{h}_i^K$.

Then $\hat{h}^K = (\hat{h}_1^K,\ldots,\hat{h}_p^K)$. END.

So, Algorithm MA1(L) associates the pruning rules coming from Chapter 6 and 7 with the decomposition result of Chapter 9. A computation of the measurement regions $R_i(k)$ will be done in the next chapter for the application to multitarget tracking.

A similar algorithm MA2(L) can be derived for the estimation of (s^K,A^K).

The resulting estimates $\hat{d}^K$ or $(\hat{d}^K,\hat{x}^K)$ given by the most likely (d^K,A^K) or (d^K,A^K,x^K) are not necessarily the MAP estimates of d^K and (d^K,x^K) respectively.

The continuous state estimates are computed for the most likely data hypothesis, i.e., the data association, and do not take into account all possible data hypotheses weighted by their probability. Reid [9] and Keverian [10] take the same approach in their multitarget tracking algorithm. So, in strict estimation terms, our estimates for the discrete or hybrid state sequences are optimal only in a rather unique sense. On the other hand, this different estimation problem leads to much more tractable computations. Therefore, we suggest this suboptimal scheme.

Using Algorithm MA1(L) or MA2(L), the storage requirements can still be too large with respect to our resources. In this case, the

algorithms are truncated to some manageable length K_r, in order to keep the number of hypotheses at a reasonable level. We decide upon the pair $(\hat{d}(k-K_r), \hat{A}(k-K_r))$ at time k by choosing the most likely hypothesie sequence from $k - K_r - 1$ to k. The decomposition property applies, whenever the sets $Y(k) \cap R_i(k)$ are mutually exclusive during an interval $[k - K_r, k]$, even though there has been some overlap before time $k - K_r$.

The proposed algorithm in the next chapter for solving the multitarget tracking problem is based on a MA1(L) type algorithm truncated to a length K_r.

10.3 Summary

In this chapter, an algorithm for estimating the discrete or hybrid state sequence of a multiple Gauss-Markov process has been presented. It contains both the implementable pruning techniques of Chapter 7 and the decomposition possibilities of Chapter 9.

In the next chapter, we discuss an example application to multitarget tracking.

CHAPTER 11

APPLICATION TO MULTITARGET TRACKING

The purpose of this chapter is to apply the algorithm presented in the last chapter to the multitarget case. In particular, a computation of the measurement regions $R_i(k)$ will be done, and the decomposition possibilities will be discussed in this context through a simple numerical example.

11.1 Model and Objective for the Multitarget Tracking Problem

The model for the set of targets has been presented in Chapter 2 as an example of a multiple Gauss-Markov process. We recall briefly the main features.

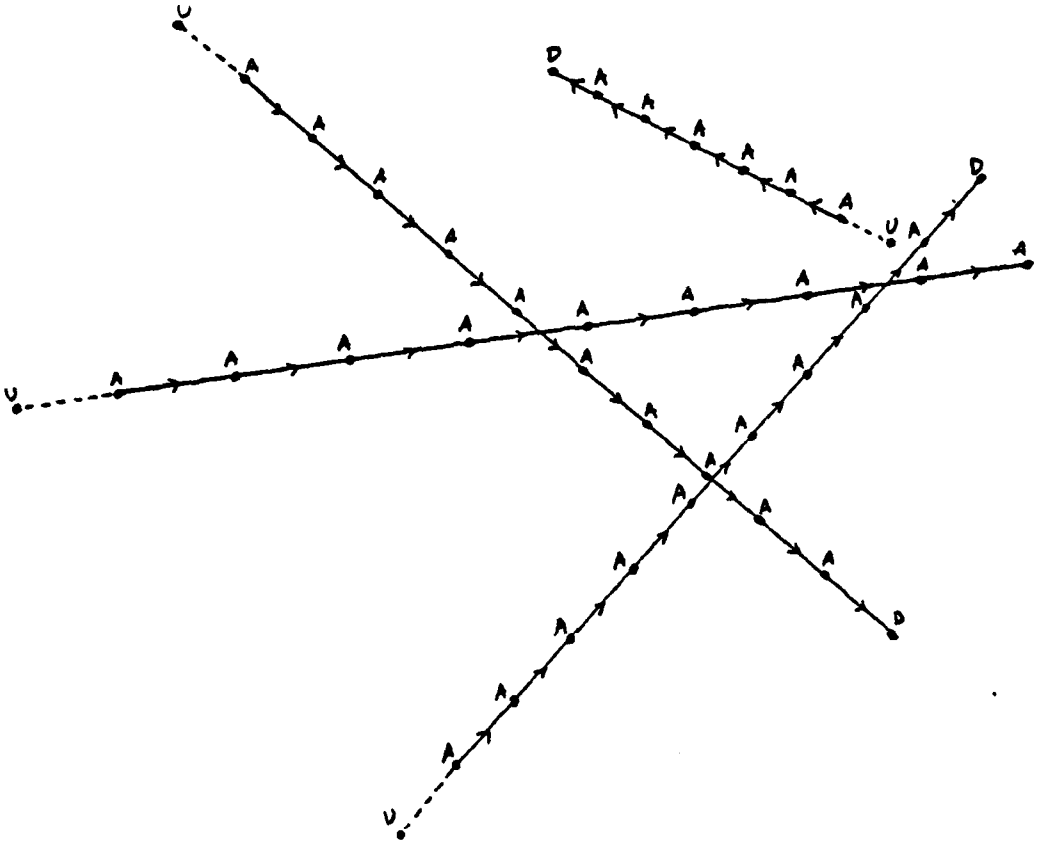
The system considered consists of p targets. We assume that each target goes successively from an unborn discrete state U, to an alive state A, to a dead state D. Planar, straight-line motions are assumed for the targets (see Fig. 11.1). So, the state $s_i(k) = \{d_i(k), x_i(k)\}$ is defined by:

$$d_i(k) \in \{U,A,D\}$$

$$x_i(k) = \begin{bmatrix} x_i^1(k) \\ v_i^1(k) \\ x_i^2(k) \\ v_i^2(k) \end{bmatrix} \quad \text{if } d_i(k) = A$$

Each target i can be detected by the surveillance system with a con-

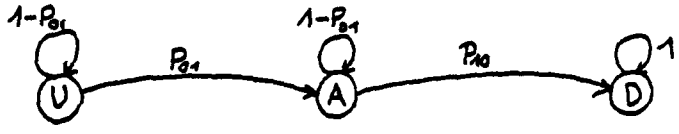
FIGURE 11.1



SET OF TARGETS

stant probability of detection P_d. The probability for a false alarm is a constant P_f. We assume that the detectability of each target is independent of the others and that the process generating a false alarm is independent of the other false alarms. We assume also independent target dynamics. So, we have for each target i:

- the discrete dynamics:



- the continuous dynamics: if $d_i(k) = d_i(k+1) = A$

$$x_i(k+1) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i(k) + w_i(k) \quad , \quad w_i(k) \sim N(0,Q_i)$$

$$x_i(0) \sim N(0,P_i^0)$$

- the measurement equation: if $d_i(k) = A$,

$$z_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_i(k) + v_i(k) \quad , \quad v_i(k) \sim N(0,R_i);$$

if $d_i(k) \in \{U,D\}$, no measurement is generated; so, we can set $z_i(k)$ to 0.

- the false alarms distribution:

$$z_f(k) \sim N(0,\Sigma) \quad , \quad \Sigma > 0$$

- the data hypothesis statistics:

$$P(A(k)|d(k)) = P_d^{M_d(k)} \times (1-P_d)^{M_m(k)} \times P_f^{M_f(k)} \quad , \text{ where:}$$

$M_d(k)$ = # detections under A(k)

$M_m(k)$ = # missed detections under A(k)

$M_f(k)$ = # false alarms under A(k) .

The purpose of the surveillance system is to decide upon the discrete states of the targets, and to estimate their locations and velocities, with a maximum delay K_r, while minimizing the probability of an error in the decisions. The delay K_r takes into account both the computational resources and the necessity for a timely decision.

Algorithm MA1(L) of the last chapter, truncated to a length K_r, applies directly to this problem, since it computes, at each time k and for each target i, the discrete state $\hat{d}_i(k - K_r)$, the data association $\hat{A}_i(k - K_r)$. The location and velocity estimates can be simply given by the continuous state estimate $\hat{x}^i_{k-K_r|k-K_r}(\hat{d}_i(k-K_r), \hat{A}_i(k-K_r))$ which is stored at time $k - K_r$ in the algorithm. Using MA1(L), we minimize the probability of an error in deciding upon the discrete states of the target and in performing the data association.

To take advantage of the decomposition possibilities in this algorithm, we must compute explicitly the measurement regions $R_i(k)$. This would show the optimality of a gating.

11.2 Optimality of a Gating

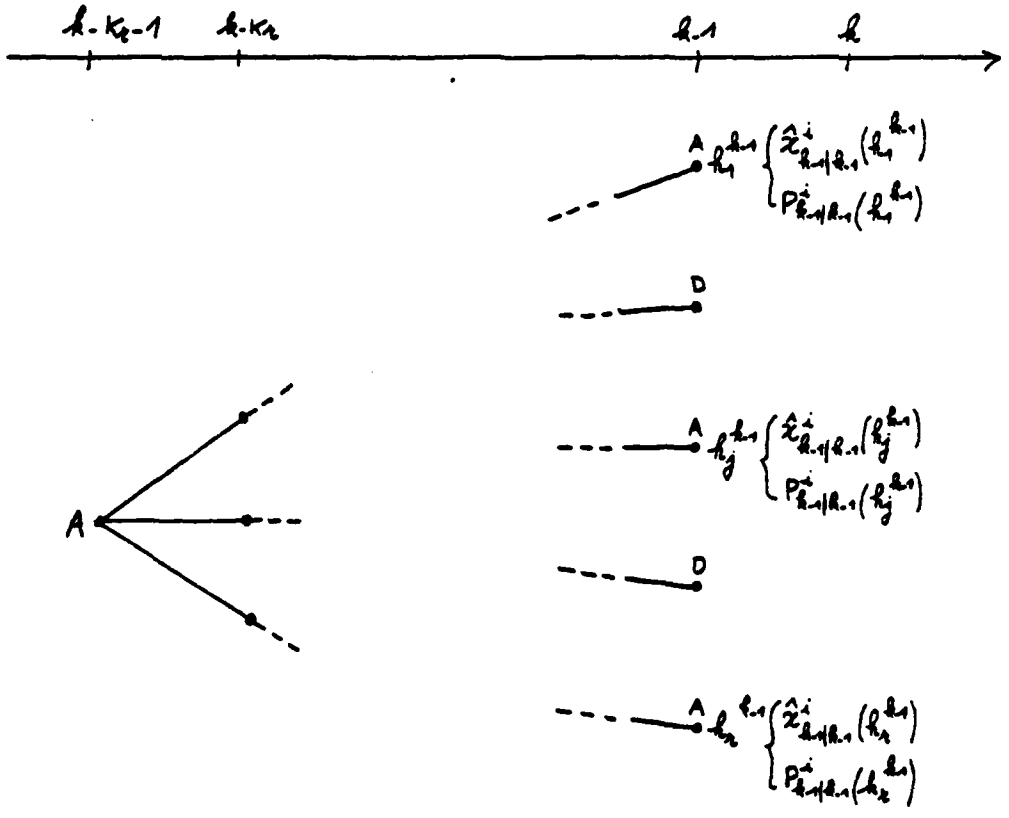
We consider a given target among the set of targets, whose existence is assumed on at least one remaining hypothesis in the tree at time k-1. For each such hypothesis, we have an estimate of the position

and velocity of the target and the corresponding covariance (see Fig. 11.2). So, the predicted values for the next measurements coming from that target and their covariances can be computed. If these covariances are small and we receive at the next instant of time a measurement far away from these predicted values, it is unlikely that it should be associated with this target, since it would require a very unlikely value of the measurement noise. Rather, we would suspect that this measurement is a false alarm or is due to another target. This intuitive approach is taken in [9]-[10], where an arbitrary gate is drawn around the predicted value of the measurement for each hypothesis; a measurement falling outside this gate is not associated with the target. A justification for this gating can be found in our optimal hypothesis deletion techniques as follows.

We have to look for a hypothesis at time k, that consider the received measurement z(k) as a false alarm, and could delete, using R1(L), the measurement association of z(k) to the target, for some z(k). These values of z(k) define a region outside which no measurement association is to be made to the target. We can think of simply extending each hypothesis sequence on the target at time k-1 by the two following hypotheses: the target is still alive and the measurement z(k) is assigned to it, or the target is alive but z(k) is a false alarm (see Fig. 11.3). Unfortunately, it can be shown that there exists no measurement for which the first extended hypothesis can be deleted by the second extended hypothesis, using R1(L): the corresponding comparison coefficient β is equal to:

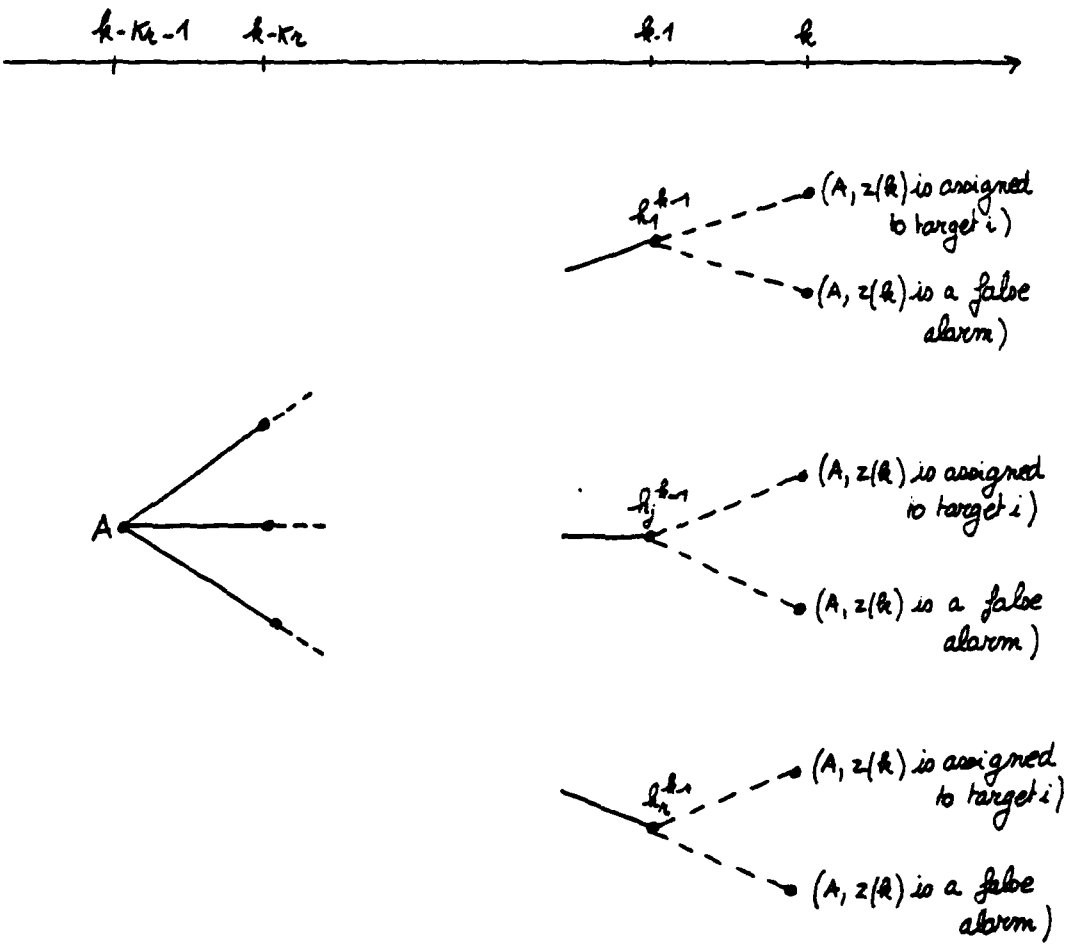
$$z^T(k)\Sigma^{-1}z(k) + 2\,\ell n\frac{P_d}{(1-P_d)P_f} + \ell n\,\frac{|\Sigma|}{|R|} \quad ,$$

FIGURE 11.2



TREE FOR TARGET i AT TIME $k-1$

FIGURE 11.3


k-K_{k-1}
k-K_k
k-1
k
A
(A, z(k) is assigned to target i)
(A, z(k) is a false alarm)
(A, z(k) is assigned to target i)
(A, z(k) is a false alarm)
(A, z(k) is assigned to target i)
(A, z(k) is a false alarm)

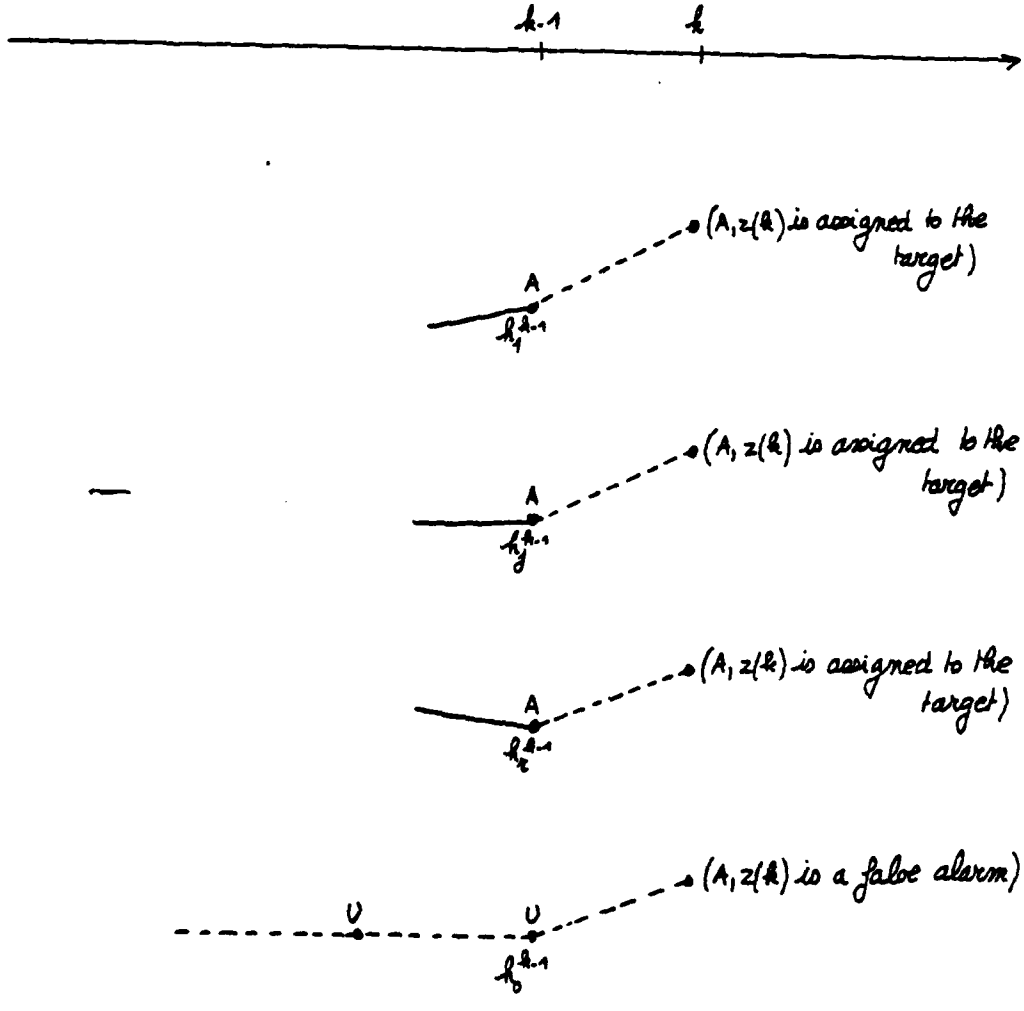
which is positive, for all z(k).

Instead, we must try to compare each hypothesis at time k that associates a measurement z(k) with the target against another hypothesis: that a secondary target is just born at time k but not detected and z(k) is a false alarm. For this alternative hypothesis, we assume only a priori information on the secondary target state: an a priori mean and an a priori (large) covariance, i.e., no information at all (see Fig. 11.4). This hypothesis implies that the preceeding measurements associated with the primary target are now considered as false alarms. If we assume that once in the past the existence of the primary target has been proved, because all the survivors went to the alive state A, then the alternative hypothesis is only considered for our gating; we do not have to take into account the hypothesis that the secondary target is just born and that a measurement is associated to it. This argument applies to the single object case as well.

It is possible to determine the set of measurements z(k) for which all the hypotheses that would associate the observation can be deleted by that last alternative hypothesis. This corresponds to our measurement region $R_i(k)$ for target i at time k. An actual computation of such a region proves the existence of an optimal gating. It can be done in the following way.

Let us consider a specific target, the primary target, at time k-1, and the corresponding remaining hypothesis sequences $\mathcal{H}^{k-1} = \{h_j^{k-1}, 1 \leq j \leq r\}$ (the index i of the target is omitted for simplicity in the following notations) terminating at $d(k-1) = A$. For all $h_j^{k-1} \in \mathcal{H}^{k-1}$, we have an estimate $\hat{x}_{k-1|k-1}(h_j^{k-1})$ and a covariance $P_{k-1|k-1}(h_j^{k-1})$.

FIGURE 11.4


k-1
k
A
(A, z(k) is assigned to the target)
A
(A, z(k) is assigned to the target)
A
(A, z(k) is assigned to the target)
U
U
(A, z(k) is a false alarm)

We can compute the predicted value $\hat{z}_j(k)$ of the measurement, corresponding to h_j^{k-1}, from $\hat{z}_j(k) = H\hat{x}_{k|k-1}(h_j^k)$, and the covariance $B(h_j^k) = HP_{k|k-1}(h_j^k)H^T + R$, for all $h_j^k = (h_j^{k-1}, h_d(k))$ where $h_d(k)$ is the hypothesis (d(k) = A, the measurement originates from the primary target). Under our model assumptions, we have:

$$P(h_j^k, Y^k) = P(h_j^{k-1}, Y^{k-1}) \times (1 - P_{10}) \times P_d \times N[z(k) - \hat{z}_j(k), B(h_j^k)] \tag{11.1}$$

$\hat{x}_{k|k}(h_j^k)$, $P_{k|k}(h_j^k)$ are updated with z(k) using the Kalman Filtering Equations (7.2)-(7.7).

We compare each hypothesis h_j^k with the alternative hypothesis h_0^k = (the secondary target is just born at time k, not detected and z(k) is a false alarm). We have:

$$P(h_0^k, Y^{k-1}) = P(h_0^{k-1}, Y^{k-1}) \times P_{01} \times (1 - P_d) \times P_f \times N[z(k), \Sigma] \tag{11.2}$$

$\hat{x}_{k|k}(h_0^k) = x_0$, $P_{k|k}(h_0^k) = P_0$ is the a priori "large" covariance.

We have $P_{k|k}(h_j^k) < P_0$, since under h_j^k we have updated the estimate and covariance at least once. So, according to R1(L), we don't assign z(k) to the target if the corresponding $\beta(h_0^k, h_j^k)$ is less or equal than zero. We denote by $R(h_j^{k-1})$ the set of measurements of the space Z for which $\beta(h_0^k, h_j^k) > 0$, i.e.,

$$R(h_j^{k-1}) = \{z(k) \in Z \text{ s.t. } \beta(h_0^k, h_j^k) > 0\}$$

Then, the measurement region R(k) corresponding to the target is simply given by:

$$R(k) = \bigcup_{k\_j \le r} R(h_j^{k-1})$$

In Appendix D, we show that,

$$z(k) \in R(h_j^{k-1}) \iff \beta(h_0^k, h_j^k) > 0$$

$$\iff (z(k) - m_j(k))^T E_j(k) (z(k) - m_j(k)) < \varepsilon_j(k) \tag{11.3}$$

where

$$E_j(k) = (R + H[P^{-1}_{k|k-1}(h_j^k) - P_0^{-1}]^{-1} H^T)^{-1} - \Sigma^{-1} \tag{11.4}$$

$$m_j(k) = E_j^{-1}(k)(R + H[P^{-1}_{k|k-1}(h_j^k) - P_0^{-1}]^{-1} H^T)^{-1} H c_j(k) \tag{11.5}$$

$$c_j(k) = \hat{x}_{k|k-1}(h_j^k) + P_{k|k-1}(h_j^k)[P_0 - P_{k|k-1}(h_j^k)]^{-1}(\hat{x}_{k|k-1}(h_j^k) - x_0) \tag{11.6}$$

$$\varepsilon_j(k) = (Hc_j(k))^T [\Sigma - R - H(P^{-1}_{k|k-1}(h_j^k) - P_0^{-1})^{-1} H^T]^{-1} (Hc_j(k)) + \delta_j(k) + 2 \ln \frac{P_d}{(1-P_d)P_f} + 2 \ln \frac{(1-P_{10})}{P_{01}} + \ln \frac{|\Sigma|}{|R|} \tag{11.7}$$

$$\delta_j(k) = (\hat{x}_{k|k-1}(h_j^k) - x_0)^T (P_0 - P_{k|k-1}(h_j^k))^{-1} (\hat{x}_{k|k-1}(h_j^k) - x_0) + 2 \ln \frac{P(h_j^{k-1}, Y^{k-1})}{P(h_0^{k-1}, Y^{k-1})} + \ln \frac{|P_0|}{|P_{k|k-1}(h_j^k)|} . \tag{11.8}$$

We can assume that: $P_d > (1-P_d)P_f$, $\Sigma >> R$, $P_{01} + P_{10} < 1$; $\delta_j(k)$ is positive, since it corresponds to $\beta(h_j^k, h_0'^k)$ where $h_0'^k = (h_0^{k-1}, h_d(k))$ (if $\beta(h_j^k, h_0'^k) \le 0$, then we could immediately discard h_j^k, whatever $z(k)$ is). So, if $H[P^{-1}_{k|k-1}(h_j^k) - P_0^{-1}]^{-1} H^T < \Sigma - R$ (11.9), then $E_j(k) > 0$ and

$\varepsilon_j(k) \geq 0$. Condition (11.9) is satisfied, if Σ is much larger than R and $P_{k|k-1}(h_j^k)$ much smaller than P_0, which is generally the case.

In this case, $R(h_j^{k-1})$ defines an ellipse centered at $m_j(k)$. The measurement region R(k) for this target is the union of several ellipses $R(h_j^{k-1})$, whose computations are quite simply using (11.3)-(11.8). An approximation of R(k) without loss of estimation performance is given by one ellipse circumscribing all others (see Fig . 11.5). Thus, all the measurements falling outside this region R(k) won't be associated with the target. This shows the existence of an optimal gating!

A special case of interest is, when the distribution of the false alarms tends to a uniform distribution, i.e., $\Sigma \to \infty$, and P_0 is a very large initial covariance, so that we can assume $P_{k|k-1}(h_j^k) << P_0$ for all $h_j^k \in \mathcal{H}^{k-1} \times h_d(k)$. In this case, the parameters $m_j(k)$, $E_j(k)$, $\varepsilon_j(k)$ of the ellipse take the following simple form:

$$m_j(k) \simeq H\hat{x}_{k|k-1}(h_j^k) = \hat{z}_j(k) \tag{11.10}$$

$$E_j(k) \simeq (R + HP_{k|k-1}(h_j^k)H^T)^{-1} = [B(h_j^k)]^{-1} \tag{11.11}$$

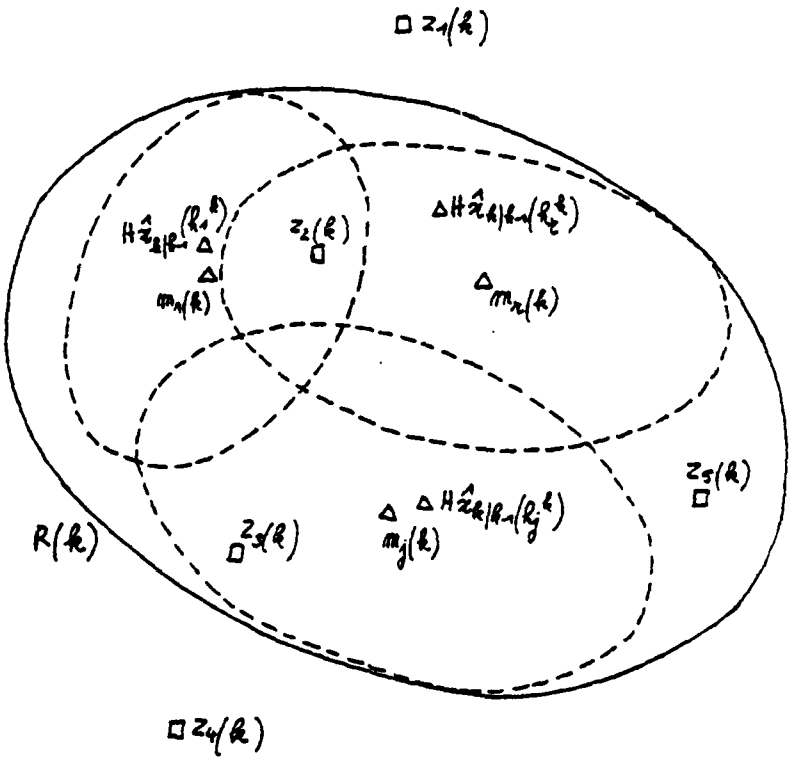
$$\varepsilon_j(k) \simeq 2\,\ell n \frac{P_d}{(1-P_d)P_f} + 2\,\ell n \frac{1-P_D}{P_{01}} + \ell n \frac{|\Sigma|}{|R|} + \delta_j(k)$$

$$+ (H\hat{x}_{k|k-1}(h_j^k))^T\Sigma^{-1}(H\hat{x}_{k|k-1}(h_j^k)) \tag{11.12}$$

The ellipse $R(h_j^{k-1})$ becomes centered at the predicted value of the measurement $\hat{z}_j(k)$.

The previous algorithms of Reid [9] and Keverian [10] use a gating around the predicted value of the form: $(z(k) - \hat{z}_j(k))^T(B(h_j^k))^{-1}(z(k) - \hat{z}_j(k)) < \eta$, where η is an arbitrary threshold. Using the above result, we now have a mean of determining the optimal threshold equal to $\varepsilon_j(k)$ defined by (11.12).

FIGURE 11.5



OPTIMAL GATING at TIME k for TARGET i

△ : center of an ellipse = predicted value of the measurement

□ : actual measurements

So, for each target i at time k-1, we compute the ellipses $R_i(k)$ using (11.4)-(11.8). If, for each k, the sets $Y(k) \cap R_i(k)$ are mutually exclusive, then our multitarget problem is equivalent to p independent single target problems.

We now look at a simple numerical example to illustrate the applicability of this optimal gating.

11.3 Example

The following example focuses on the decomposition possibilities in the multitarget tracking algorithm, i.e., computation and application of the optimal gating.

For this purpose, we consider the multiple system as consisting of two targets. We assume that the existence of these two targets has been decided, because the single remaining survivor, or the most likely sequence in case of a truncation, went through the alive state for both targets at one time. We assume also that the covariances for the continuous state estimates have reached their steady-state values (in 5 to 10 steps with the following values). For simplicity, at each time k we compute for each target only the ellipse corresponding to the comparison of the hypothesis sequence $h^k_{A,i}$ (primary target i always detected and a measurement is assigned to it) with the hypothesis sequence $h^k_{0,i}$ (secondary target i is just born at time k, but not detected and the measurement is a false alarm). We study the variation in the size of this ellipse as a function of the process and measurement noises and the efficiency of the gating as these two targets are crossing.

We consider Q_i and R_i matrices of the form: for i = 1,2,

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q \end{bmatrix}, \quad R_i = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} ,$$

where the process noise enters only the velocity components.

We take the following values for the other parameters:

- discrete dynamics: $P_{01} = P_{10} = 0.1$
- initial conditions:

$$x_0^i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad P_0^i = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \text{ for } i = 1,2$$

- detection and false alarm probabilities:

$$P_d = 0.66; \; P_f = 0.33$$

- false alarms covariance:

$$\Sigma = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}$$

Assume that at time k-1, we have the following probabilities and estimates for target 1 and 2:

$$P(h_{A,1}^{k-1}|Y^{k-1}) = P(h_{A,2}^{k-1}|Y^{k-1}) = 0.5$$

$$P(h_{0,1}^{k-1}|Y^{k-1}) = P(h_{0,2}^{k-1}|Y^{k-1}) = 0.01$$

$$\hat{x}^1_{k|k-1}(h_{A,1}^{k-1}) = \begin{bmatrix} -20 \\ +5 \\ -20 \\ +5 \end{bmatrix} \quad \hat{x}^2_{k|k-1}(h_{A,2}^{k-1}) = \begin{bmatrix} +20 \\ -5 \\ -20 \\ +5 \end{bmatrix}$$

We compute the ellipses (in fact circles) for different values of q and r. Since we take $r << 500$ and we have $P^i_{k|k-1}(h^k_{A,i}) << P_0$, these circles are centered at the predicted values $\begin{pmatrix}-20\\-20\end{pmatrix}$ for target 1 and $\begin{pmatrix}20\\-20\end{pmatrix}$ for target 2. In Fig. 11.6 we have plotted the radius ρ of these circles as a function of q and r.

We see that for q or r greater than 1, the radius grows rapidly, so that the gating becomes useless. As r tends to 0 for q equal to 1, this radius tends slowly to $+\infty$, because of the term $\ell n \frac{1}{|R|}$. For r=1, as q tends to 0, it tends also slowly to $+\infty$ because of $-\ell n|P^i_{k|k-1}(h^k_{A,i})|$ in $\delta_j(k)$.

Thus, we see that the size of the gate is relatively insensitive to noise modelling errors for small values of q and r (less than 0.1 in our example).

Now for $q = r = 1$, we study the efficiency of the gating as two targets are crossing. To limit the number of hypotheses, and subsequently the number of circles, we assume $K_r = 1$.

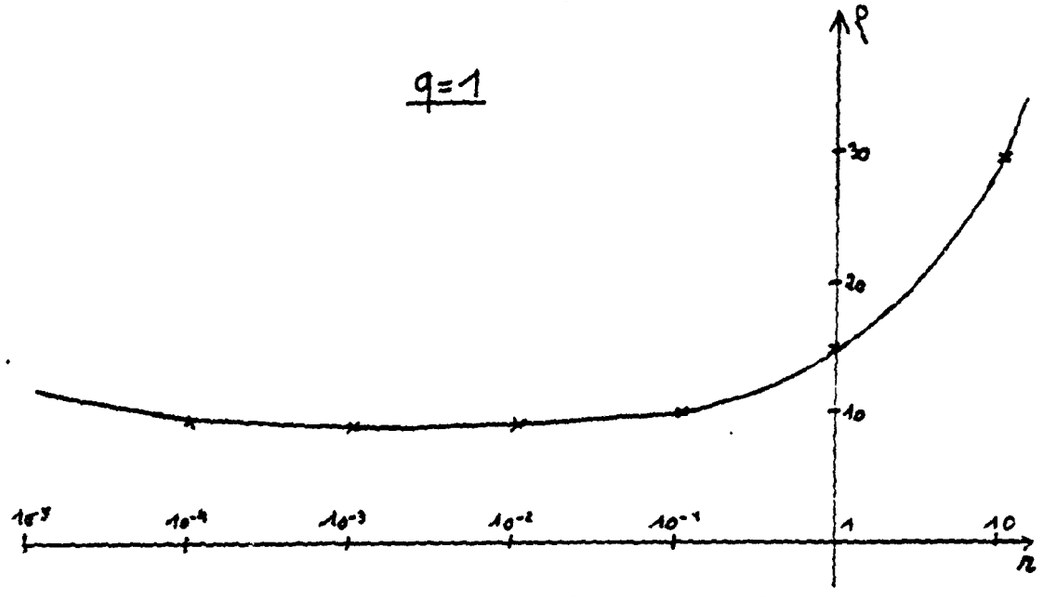
We start at time k-1 with the above estimates $\hat{x}^1_{k|k-1}(h^k_{A,1})$ and $\hat{x}^2_{k|k-1}(h^k_{A,2})$ and hypothesis probabilities for target 1 and 2. We receive some sets of measurements Y(k),Y(k+1),Y(k+2),..., and each for each time k, k+1, k+2,..., we compute the optimal gates for target 1 and target 2 and apply the decomposition theorem, whenever this is possible (see Fig. 11.7-11.9):

<u>step k</u>: $Y(k) \cap R_1(k) \cap R_2(k) = \emptyset$ (figure 11-7a)

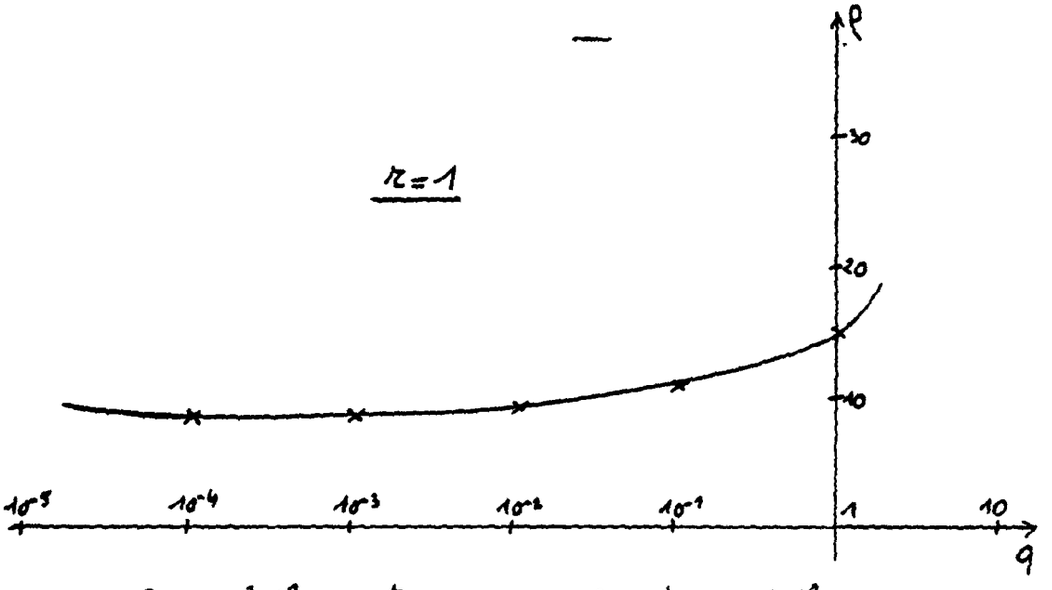
Assign $z_1(k)$ to target 1, $z_2(k)$ to target 2,

$z_3(k)$, $z_4(k)$ to the set of false alarms.

FIGURE 11.6



Size of the gate as a function of the measurement noise



Size of the gate as a function of the process noise

step k+1: $Y(k+1) \cap R_1(k+1) \cap R_2(k+1) = \emptyset$ (figure 11-7b)

Assign $z_1(k+1)$ to target 1, $z_2(k+1)$ to target 2, $z_3(k+1)$ to the set of false alarms

step k+2: $Y(k+2) \cap R_2(k+2) \cap R_1(k+2) \neq \emptyset$ (figure 11-8a)

Data association uncertainty for $z_1(k+2), z_2(k+2)$.

Assign $z_3(k+2), z_4(k+2)$ to the set of false alarms.

step k+3: (figure 11-8b)

Since $K_r = 1$, MAP decision rule assigns $z_2(k+2)$ to target 1 and $z_1(k+2)$ to target 2 at time k+2.

Data association uncertainty for $z_1(k+3), z_2(k+3)$.

step k+4 (figure 11-9a)

MAP decision rule assigns $z_2(k+3)$ to target 1 and $z_1(k+3)$ to target 2.

Computation of the corresponding $R_1(k+4)$, $R_2(k+4)$.

$Y(k+4) \cap R_1(k+4) \cap R_2(k+4) = \emptyset$ Assign $z_1(k+4)$ to target 1, $z_2(k+4)$ to target 2, $z_3(k+4), z_4(k+4), z_5(k+4)$ to the set of false alarms.

step k+5: $Y(k+5) \cap R_1(k+5) \cap R_2(k+5) = \emptyset$ (figure 11-9b)

Assign $z_2(k+5)$ to target 1, $z_1(k+5)$ to target 2, $z_3(k+5)$ to the set of false alarms.

We see that the crossing of the two targets introduces a correlation at time k+2, k+3, k+4, but, after that time the decomposition

property is regained. Therefore, we must run one algorithm for both targets only for steps k+2, k+3, k+4.

This example highlights the efficiency of our optimal gating in a very simple case, where, for simplicity, we compute only the gates corresponding to the targets being detected and we take the delay K_r equal to one.

Complete simulations would have to be run to show the real capabilities of our optimal pruning rules in Algorithm MA1(L), and the efficiency of our gating in more complex situations, where we have many targets in the area, we take a delay greater than one to decide upon the discrete state of the targets and the data associations, and we consider all hypotheses for each target.

11.4 Summary

In this chapter, we have computed an optimal gating for the multitarget tracking problem. We have shown that, under some approximations, this gating took a form similar to the one in the heuristic procedure of Reid's algorithm [9]. Then we have derived a simple example, that illustrates its sensitivity with respect to the noise parameters, and its efficiency as two targets are crossing. These encouraging results are only the first stage in an evaluation of the performance of our multitarget tracking algorithm.

FIGURE 11.7

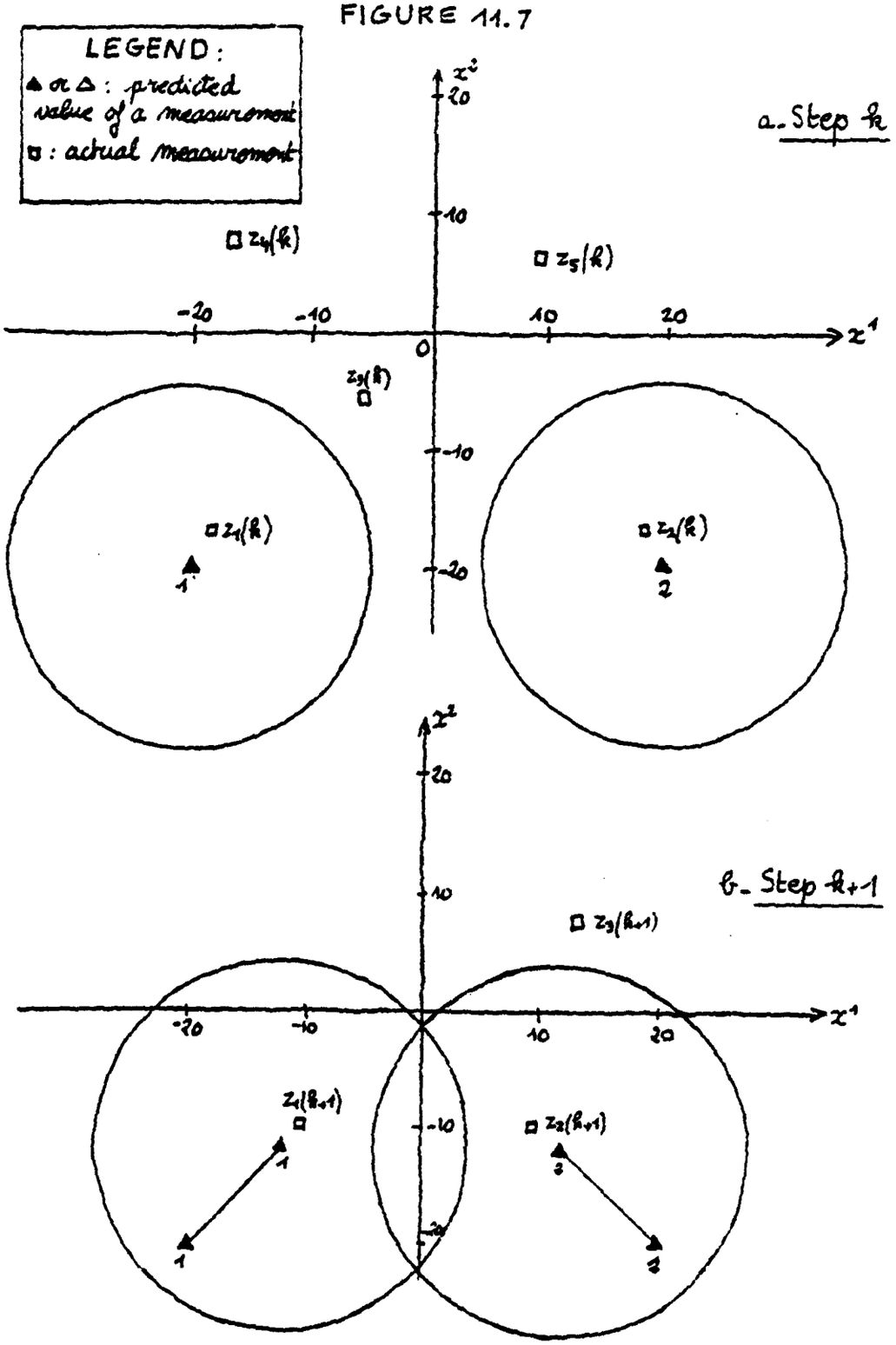
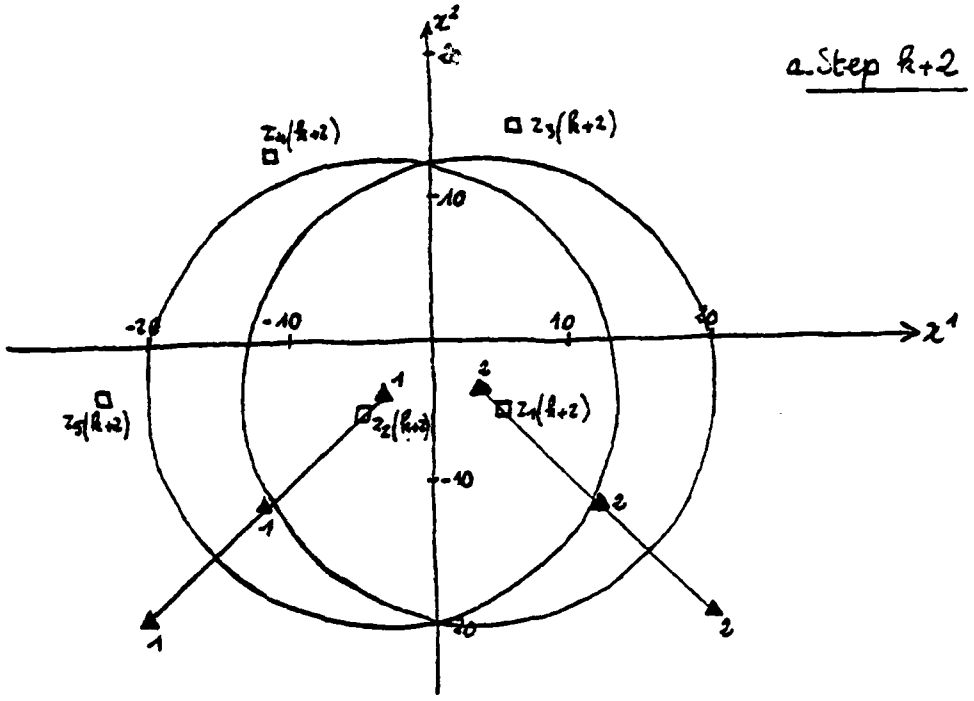

LEGEND :
▲ or △ : predicted value of a measurement
□ : actual measurement
a- Step k
z^2
20
10
$z_4(k)$
$z_5(k)$
-20
-10
10
20
z^1
0
$z_3(k)$
-10
$z_1(k)$
$z_2(k)$
-20
1
2
b- Step k+1
$z_3(k+1)$
$z_1(k+1)$
$z_2(k+1)$

FIGURE 11.8


a. Step k+2
z²
z¹
z₄(k+2)
z₃(k+2)
z₅(k+2)
z₂(k+2)
z₁(k+2)
10
-10
-20
20
1
2

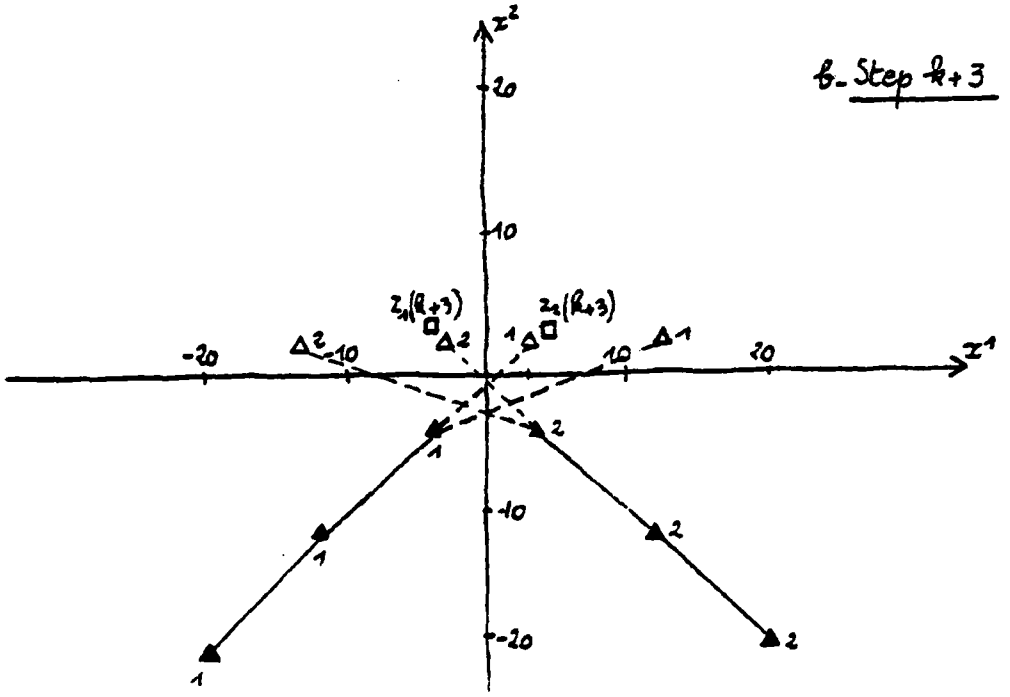
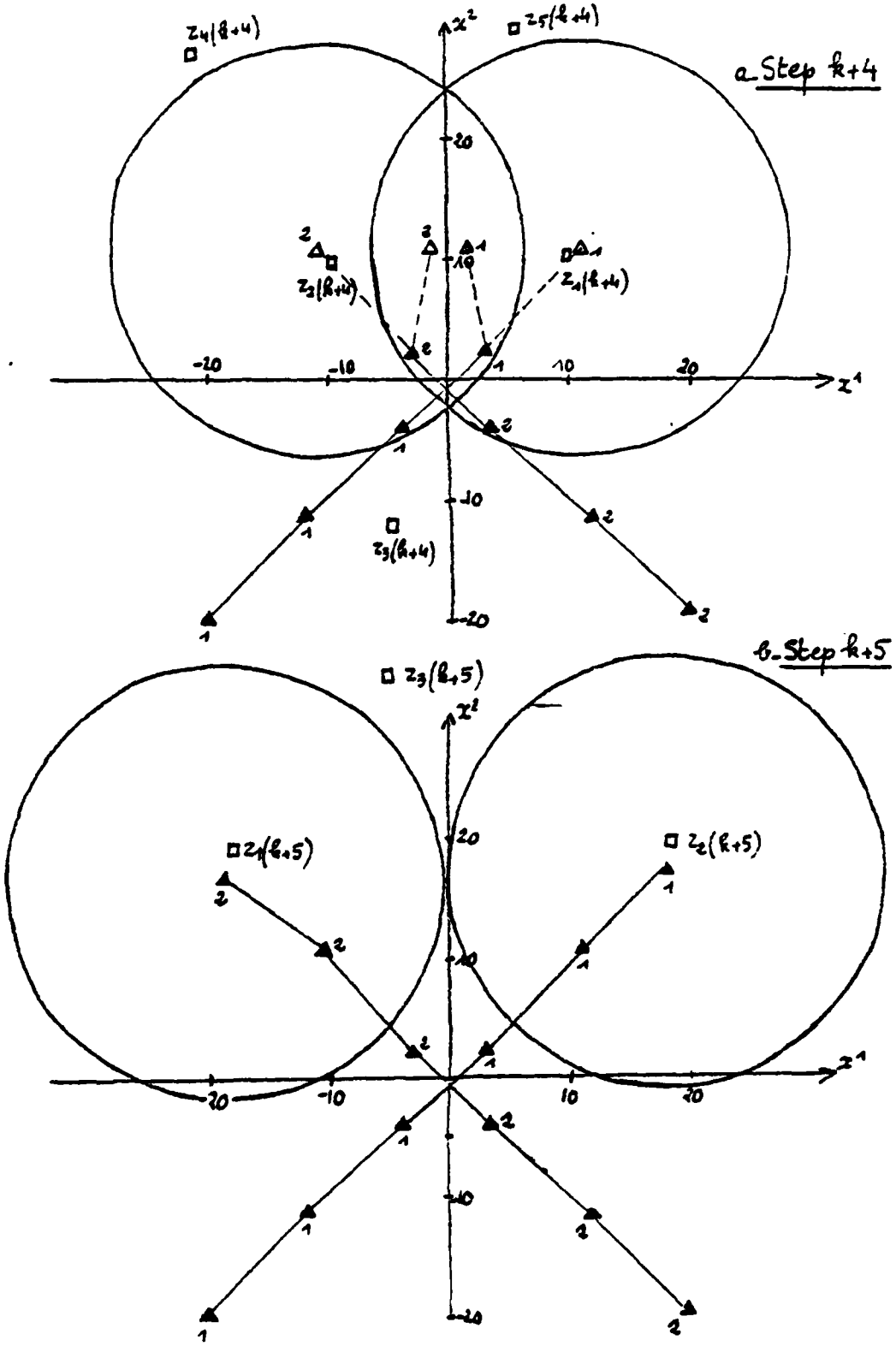

b. Step k+3
z²
z¹
z₂(k+3)
z₁(k+3)
20
10
-10
-20
1
2

FIGURE 11.9


a. Step k+4
z4(k+4)
z5(k+4)
z2(k+4)
z1(k+4)
z3(k+4)
x2
x1
20
10
-10
-20
b. Step k+5
z3(k+5)
z1(k+5)
z2(k+5)

PART IV

CONCLUSION

CHAPTER 12

CONCLUSION AND FURTHER WORK

The purpose of this thesis was to set up a common framework for different hybrid state estimation problems, and to present optimal algorithms for estimating the state sequence of single and multiple Gauss-Markov processes. We derived optimal techniques for reducing the amount of computation, and, in the multiple system case, we proved that clustering can be optimal under certain conditions. We also computed an optimal gating, which decreases the computational complexity.

Simple applications to a failure detection problem and a multi-target tracking problem showed encouraging results. Due to the lack of time, we were not able to perform more complete simulations in order to evaluate the real performances of our algorithms: efficiency of the pruning rules and the gating, sensitivity to the various parameters.

Our hope is that this thesis will provide a basis for rational (if not optimal) procedures applicable to diverse hybrid state estimation problems.

A further area of research would be to apply this approach to a distributed system, where one single decision maker cannot handle completely the estimation of the hybrid system, so that this task must be divided among several local decision makers, that limit the communications between them at the minimum level required to insure a coordination in their decisions.

APPENDIX A

Some Useful Matrix Identities

Let B_1 and B_2 be two $n \times n$ square invertible matrices. We have the following equalities:

$$B_1^{-1} - B_1^{-1}(B_1^{-1} - B_2^{-1})^{-1}B_1^{-1} = (B_1 - B_2)^{-1} \tag{A.1}$$

$$B_2^{-1} + B_2^{-1}(B_1^{-1} - B_2^{-1})^{-1}B_2^{-1} = (B_2 - B_1)^{-1} \tag{A.2}$$

In Jazwinsky [4], we have the following equalities:

Let P,R,M be $n \times n$, $m \times m$, $m \times n$ matrices where $P = P^T \geq 0$ and $R = R^T > 0$. Then:

$$(I + PM^TR^{-1}M)^{-1} = I - PM^T(MPM^T + R)^{-1}M \tag{A.3}$$

$$(I + PM^TR^{-1}M)^{-1}P = P - PM^T(MPM^T + R)^{-1}MP \tag{A.4}$$

$$(I + PM^TR^{-1}M)^{-1}PM^TR = PM^T(MPM^T + R)^{-1} \tag{A.5}$$

APPENDIX B

Proof of the Lemma in Chapter 7

$$\forall x \in X,\ f_1(x) \leq f_2(x) \tag{B.1}$$

$$\Leftrightarrow \inf_{x \in X} 2(\ln f_2(x) - \ln f_1(x)) \geq 0$$

$$\Leftrightarrow \inf_{x \in X} [x^T(B_1^{-1} - B_2^{-1})x - 2(x_1^T B_1^{-1} - x_2^T B_2^{-1})x + x_1^T B_1^{-1} x_1 - x_2^T B_2^{-1} x_2 + 2 \ln \frac{\alpha_2}{\alpha_1}] \geq 0$$

The extremum is equal to: $x_0 = (B_2^{-1} - B_1^{-1})^{-1}(B_1^{-1}x_1 - B_2^{-1}x_2)$.

It is a minimum if $B_1 < B_2$.

Therefore:

$$(B.1) \Leftrightarrow \begin{cases} B_1 < B_2 \\ x_1^T B_1^{-1} x_1 - x_2^T B_2^{-1} x_2 - (x_1^T B_1^{-1} - x_2^T B_2^{-1})(B_2^{-1} - B_1^{-1})^{-1}(B_1^{-1}x_1 - B_2^{-1}x_2) + 2 \ln \frac{\alpha_2}{\alpha_1} \geq 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} B_1 < B_2 \\ 2 \ln \frac{\alpha_2}{\alpha_1} + x_1^T[B_1^{-1} - B_1^{-1}(B_1^{-1} - B_2^{-1})^{-1}B_1^{-1}]x_1 + x_2^T[-B_2^{-1} - B_2^{-1}(B_1^{-1} - B_2^{-1})^{-1}B_2^{-1}]x_2 + 2x_2^T B_2^{-1}(B_1^{-1} - B_2^{-1})B_1^{-1}x_1 \geq 0 \end{cases}$$

Using (A.1)-(A.2):

$$(B.1) \Leftrightarrow \begin{cases} 2 \ln \frac{\alpha_2}{\alpha_1} + x_1^T(B_1 - B_2)^{-1}x_1 + x_2^T(B_1 - B_2)^{-1}x_2 - 2x_2^T(B_2 - B_1)^{-1}x_1 \geq 0 \\ B_1 < B_2 \end{cases}$$

$$\Leftrightarrow \begin{cases} B_1 < B_2 \\ (x_1 - x_2)^T (B_2 - B_1)^{-1} (x_1 - x_2) \leq 2 \ln \frac{\alpha_2}{\alpha_1} \end{cases}$$

Q.E.D.

APPENDIX C

Proof of relations (7.15)-(7.16) on β,γ

The proof is straightforward using some algebraic manipulations.

Let d_1^{k+1}, d_2^{k+1} be 2 state sequences such that $d_1(k) = d_2(k)$ and $d_1(k+1) = d_2(k+1)$. Let $\beta(d_1^k,d_2^k)$, $\beta(d_1^{k+1},d_2^{k+1})$, $\gamma(d_1^k,d_2^k)$, $\gamma(d_1^{k+1},d_2^{k+1})$ be the corresponding coefficients as defined by (7.13)-(7.14).

Using (7.2)-(7.7) and (A.3)-(A.5) of Appendix A, we have for i = 1,2:

$$\hat{x}_{k+1|k+1}(d_i^{k+1}) = (I + P_{k+1|k}(d_i^{k+1})H^T(q,k+1)R^{-1}(q,k+1)H(q,k+1))^{-1} \times x(\hat{x}_{k+1|k}(d_i^{k+1}) + P_{k+1|k}(d_i^{k+1})H^T(q,k+1)R^{-1}(q,k+1)z(k+1)) \tag{C.1}$$

$$P^{-1}_{k+1|k+1}(d_i^{k+1}) = P^{-1}_{k+1|k}(d_i^{k+1}) + H^T(q,k+1)R^{-1}(q,k+1)H(q,k+1) \tag{C.2}$$

And using (7.8)-(7.10):

$$2\,\ell n\frac{P(d_1^{k+1}|z^{k+1})}{P(d_2^{k+1}|z^{k+1})} = \ell n\,\frac{|B(d_2^{k+1})|}{|B(d_1^{k+1})|} + (z(k+1)-H(q,k+1)\hat{x}_{k+1|k}(d_2^{k+1}))^T B(d_2^{k+1})^{-1}(z(k+1)-H(q,k+1)\hat{x}_{k+1|k}(d_2^{k+1})) - (z(k+1)-H(q,k+1)\hat{x}_{k+1|k}(d_1^{k+1}))^T B(d_1^{k+1})^{-1}(z(k+1)-H(q,k+1)\hat{x}_{k+1|k}(d_1^{k+1})) \tag{C.3}$$

where for i = 1,2, $B(d_i^{k+1}) = H(q,k+1)P_{k+1|k}(d_i^{k+1})H^T(q,k+1) + R(q,k+1)$.

Using (C.1)-(C.2)-(C.3), after some tedious calculations, it can be shown that:

$$(\hat{x}_{k+1|k+1}(d_2^{k+1}) - \hat{x}_{k+1|k+1}(d_1^{k+1}))^T (P_{k+1|k+1}(d_2^{k+1}) -$$

$$P_{k+1|k+1}(d_1^{k+1}))^{-1} (\hat{x}_{k+1|k+1}(d_2^{k+1}) - \hat{x}_{k+1|k+1}(d_1^{k+1})) +$$

$$2 \ln \frac{P(d_1^{k+1}|z^{k+1})}{P(d_2^{k+1}|z^{k+1})}$$

$$= (\hat{x}_{k+1|k}(d_2^{k+1}) - \hat{x}_{k+1|k}(d_1^{k+1}))^T (P_{k+1|k}(d_2^{k+1}) -$$

$$P_{k+1|k}(d_1^{k+1}))^{-1} (\hat{x}_{k+1|k}(d_2^{k+1}) - \hat{x}_{k+1|k}(d_1^{k+1})) +$$

$$2 \ln \frac{P(d_1^k|z^k)}{P(d_2^k|z^k)} + \ln \frac{|B(d_2^{k+1})|}{|B(d_1^{k+1})|}$$

Assuming F is invertible, it is easy to see using (7.3)-(7.5) that the above term is equal to:

$$(\hat{x}_{k|k}(d_2^k) - \hat{x}_{k|k}(d_1^k))^T (P_{k|k}(d_2^k) - P_{k|k}(d_1^k))^{-1} (\hat{x}_{k|k}(d_2^k) -$$

$$\hat{x}_{k|k}(d_1^k)) + 2 \ln \frac{P(d_1^k|z^k)}{P(d_2^k|z^k)} + \ln \frac{|B(d_2^{k+1})|}{|B(d_1^{k+1})|} \quad .$$

Finally, taking expressions (7.13)-(7.14) for $\beta(d_1^{k+1}, d_2^{k+1})$, $\beta(d_1^k, d_2^k)$, $\gamma(d_1^{k+1}, d_2^{k+1})$, $\gamma(d_1^k, d_2^k)$, the above equality implies directly conditions (7.15)-(7.16). Q.E.D.

APPENDIX D

Determination of the Optimal Gating

Let h_j^k and h_0^k be the two hypotheses as defined in Chapter 11 and $\beta(h_j^k, h_0^k)$ be the corresponding coefficient. The optimal gating is determined by $z(k) \in Z$ such that $\beta(h_j^k, h_0^k) \geq 0$. According to Chapter 11, we have:

$$\hat{x}_{k|k}(h_j^k) = \hat{x}_{k|k-1}(h_j^k) + K_k(h_j^k)[z(k) - H\hat{x}_{k|k-1}(h_j^k)] \tag{D.1}$$

$$\hat{x}_{k|k}(h_0^k) = x_0 \tag{D.2}$$

$$P_{k|k}^{-1}(h_j^k) = P_{k|k-1}^{-1}(h_j^k) + H^T R^{-1} H \tag{D.3}$$

$$P_{k|k}(h_0^k) = P_0 \tag{D.4}$$

$$P(h_j^k, \overline{Y^k}) = P(h_j^{k-1}, Y^{k-1}) \times \overline{P_d} \times (1 - P_{10}) \times N[z(k) - H\hat{x}_{k|k-1}(h_j^k), B(h_j^k)] \tag{D.5}$$

$$P(h_0^k, Y^k) = P(h_0^{k-1}, Y^{k-1}) \times P_f \times (1 - P_d) \times P_{01} \times N[z(k), \Sigma] \tag{D.6}$$

$$\beta(h_j^k, h_0^k) = (\hat{x}_{k|k}(h_j^k) - \hat{x}_{k|k}(h_0^k))^T (P_{k|k}(h_0^k) - P_{k|k}(h_j^k))^{-1} (\hat{x}_{k|k}(h_j^k) - \hat{x}_{k|k}(h_0^k)) + 2 \ln \frac{P(h_j^k, Y^k)}{P(h_0^k, Y^k)} + \ln \frac{|P_{k|k}(h_0^k)|}{|P_{k|k}(h_j^k)|} \tag{D.7}$$

Using (D.1)-(D.6), we have:

$$\beta(h_j^k, h_0^k) = (z(k) - H\hat{x}_{k|k-1}(h_j^k))^T(-T_j^k)(z(k) - H\hat{x}_{k|k-1}(h_j^k))$$

$$+ 2(\hat{x}_{k|k-1}(h_j^k) - x_0)^T(P_0 - P_{k|k}(h_j^k))^{-1}K_k(h_j^k)(z(k) -$$

$$- H\hat{x}_{k|k-1}(h_j^k)) + z^T(k)\Sigma^{-1}z(k) + \alpha_j(k) \quad , \quad \text{where:}$$

$$T_j(k) = B^{-1}(h_j^k) - K_k^T(h_j^k)[P_0 - P_{k|k}(h_j^k)]^{-1}K_k(h_j^k) \quad , \qquad (D.8)$$

$$\alpha_j(k) = 2\,\ell n \frac{P(h_j^{k-1}, Y^{k-1})}{P(h_0^{k-1}, Y^{k-1})} + 2\,\ell n \frac{P_d}{(1-P_d)P_f} + 2\,\ell n \frac{(1-P_{10})}{P_{01}}$$

$$+ \ell n \frac{|\Sigma| \times |P_0|}{|B(h_j^k) \times P_{k|k}(h_j^k)|}$$

$$+ (\hat{x}_{k|k-1}(h_j^k) - x_0)^T(P_0 - P_{k|k}(h_j^k))^{-1}(\hat{x}_{k|k-1}(h_j^k) - x_0) \qquad (D.9)$$

Using the matrix identity (M.3), it can be shown that:

$$T_j(k) = (R + H[P_{k|k-1}^{-1}(h_j^k) - P_0]^{-1}H^T)^{-1} \qquad (D.10)$$

So, $\beta(h_j^k, h_0^k) = z^T(k)\Sigma^{-1}z(k) + \alpha_j(k)$

$$+ (z(k) - H\hat{x}_{k|k-1}(h_j^k) - (T_j^k)^{-1}K_k^T(h_j^k)(P_0 - P_{k|k}(h_j^k))^{-1}$$

$$(\hat{x}_{k|k-1}(h_j^k) - x_0)^T \times (-T_j^k) \times (z(k) - H\hat{x}_{k|k-1}(h_j^k) - (T_j^k)^{-1} \times$$

$$K_k^T(h_j^k)(P_0 - P_{k|k}(h_j^k))^{-1}(\hat{x}_{k|k-1}(h_j^k) - x_0) + (\hat{x}_{k|k-1}(h_j^k) - x_0)^T$$

$$(P_0 - P_{k|k}(h_j^k))^{-1}K_k(h_j^k)(T_j^k)^{-1}K_k^T(h_j^k)(P_0 - P_{k|k}(h_j^k))^{-1}(\hat{x}_{k|k-1}(h_j^k)$$

$$- x_0) \quad .$$

Using again (M.3) it can be shown that:

$$(P_0 - P_{k|k}(h_j^k))^{-1} K_k(h_j^k)(T_j^k)^{-1} K_k^T(h_j^k)(P_0 - P_{k|k}(h_j^k))^{-1} +$$

$$(P_0 - P_{k|k}(h_j^k))^{-1} = (P_0 - P_{k|k-1}(h_j^k))^{-1} \qquad (D.11)$$

$$(T_j^k)^{-1} K_k^T(h_j^k)(P_0 - P_{k|k}(h_j^k))^{-1} = HP_{k|k-1}(h_j^k)(P_0 - P_{k|k-1}(h_j^k)) \qquad (D.12)$$

and

$$|B(h_j^k)| \times |P_{k|k}(h_j^k)| = |R| \times |P_{k|k-1}(h_j^k)| \qquad (D.13)$$

Let $c_j(k) = \hat{x}_{k|k-1}(h_j^k) + P_{k|k-1}(h_j^k)[P_0 - P_{k|k-1}(h_j^k)]^{-1}$

$$(\hat{x}_{k|k-1}(h_j^k) - x_0) \qquad (D.14)$$

From (D.10)-(D.14), we have:

$$\beta(h_j^k, h_0^k) = (z(k) - Hc_j(k))^T(-T_j^k)(z(k) - Hc_j(k))$$

$$+ z^T(k)\Sigma^{-1}z(k)$$

$$+ 2\,\ell n \frac{P_d}{(1-P_d)P_f} + 2\,\ell n \frac{(1-P_{10})}{P_{01}} + \ell n \frac{|\Sigma|}{|R|} + \delta_j(k), \text{ where:}$$

$$\delta_j(k) = 2\,\ell n \frac{P(h_j^{k-1}, Y^{k-1})}{P(h_0^{k-1}, Y^{k-1})} + \ell n \frac{|P_0|}{|P_{k|k-1}(h_j^k)|}$$

$$+ (\hat{x}_{k|k-1}(h_j^k) - x_0)^T(P_0 - P_{k|k-1}(h_j^k))^{-1}(\hat{x}_{k|k-1}(h_j^k) - x_0) \qquad (D.15)$$

$$\Leftrightarrow \beta(h_j^k, h_0^k) = 2\,\ell n \frac{P_d}{(1-P_d)P_f} + 2\,\ell n \frac{(1-P_{10})}{P_{01}} + \ell n \frac{|\Sigma|}{|R|} + \delta_j(k)$$

$$+ (z(k) - (T_j^k - \Sigma^{-1})^{-1} T_j^k Hc_j(k))^T(\Sigma^{-1} - T_j^k)(z(k) -$$

$$(T_j^k - \Sigma^{-1})^{-1} T_j^k Hc_j(k)) + (Hc_j(k))^T[\Sigma - (T_j^k)^{-1}]^{-1}(Hc_j(k)),$$

Let, $E_j(k) = T_j^k - \Sigma^{-1}$ (D.16)

$$\varepsilon_j(k) = 2 \ln \frac{P_d}{(1-P_d)P_f} + 2 \ln \frac{(1-P_{10})}{P_{01}} + \ln \frac{|\Sigma|}{|R|} + \delta_j(k)$$

$$+ (Hc_j(k))^T(\Sigma - (T_j^k)^{-1})^{-1}(Hc_j(k)). \quad \text{(D.17)}$$

Then, $\beta(h_j^k, h_0^k) = \varepsilon_j(k) - (z(k) - m_j(k))^T E_j(k)(z(k) - m_j(k))$, where $E_j(k)$, $m_j(k)$, $\varepsilon_j(k)$ are defined by: (D.10), (D.14)-(D.17). Q.E.D.

REFERENCES

Estimation Theory

[1] G.D. FORNEY: "The Viterbi Algorithm," IEEE Proceedings, March 1973.

[2] R.E. KALMAN: "A new approach to Linear Filtering and Predictions Problems," Journal of Basic Engineering, March 1960.

[3] J.S. MEDITCH: "Stochastic Optimal Linear Estimation and Control," MacGraw-Hill, 1969.

[4] A.H. JAZWINSKI: "Stochastic Processes and Filtering Theory," Academic Press, New York, 1970.

Failure Detection

[5] A.S. WILLSKY: "A Survey of Design Methods for Failure Detection in Dynamic Systems," Automatica, Vol. 12, pp. 601-611, 1976.

[6] P.M. NEWBOLD nad Y.C. HO: "Detection of Changes in the Characteristics of a Gauss-Markov Process," IEEE Trans. on Aerospace, Sept. 1968.

[7] T.T. CHIEN and M.B. ADAMS: "A Sequential Failure Detection Technique and Its Application," IEEE Trans. on Aut. Control, October 1976.

Multitarget Tracking

[8] Y. BAR SHALOM: "Tracking Methods in Multitarget Environment," IEEE Trans. on Aut. Control, August 1978.

[9] D. REID: "An Algorithm for Tracking Multiple Targets," IEEE Trans. on Aut. Control, December 1978.

[10] K.M. KEVERIAN: "Multiobject Tracking by Adaptive Hypothesis Testing," B.S. Thesis LIDS M.I.T., December 1979.

[11] R.P. HUGHES: "A Distributed Multiobject Tracking Algorithm for Passive Sensor Networks," M.S. Thesis LIDS M.I.T., June 1980.

[12] R.W. SITTLER: "An Optimal Data Association Problem in Surveillance Theory," IEEE Trans. on Mil. Electronics, April 1964.

[13] J.J. STEIN and S.S. BLACKMAN: "Generalized Correlation of Multitarget Track Data," IEEE Trans. on Aero. and Electronics Systems, November 1975.

[14] C.L. MOREFIELD: "Application of 0-1 Integer Programming to Multi-target Tracking Problems," IEEE Trans. on Aut. Control, June 1977.

[15] D.L. ALSPACH: "A Gaussian Sum Approach to the Multi-target Identification Tracking Problem," Automatica, Vol. 11 pp. 285-296, May 1975.

[16] Y. BAR-SHALOM and E. TSE: "Tracking in a Cluttered Environment with Probabilistic Data Association," Proc. 4th Symposium on Non-Linear Estimation, San Diego, September 1973.

[17] Y. BAR-SHALOM: "Extension of the Probabilistic Data Association Filter to Multitarget Environment," Proc. 5th Symposium on Non-Linear Estimation, San Diego, September 1974.